E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Sistema portable para localización de objetos y comunicación mediante audio 3D para personas invidentes

Grado en Ingeniería Informática

Trabajo Fin de Grado

Marcos Rodrigo Galvez Belmonte

Asier Ruperto Marzo Pérez

Pamplona, 10 de Junio de 2020

upna Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Grado en Ingeniería Informática

Trabajo de fin de grado

SISTEMA PORTABLE PARA LOCALIZACIÓN DE OBJETOS Y COMUNICACIÓN MEDIANTE AUDIO 3D PARA PERSONAS INVIDENTES

Marcos Rodrigo GALVEZ BELMONTE

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL, INFORMÁTICA Y TELECOMUNICACIÓN

UNIVERSIDAD PÚBLICA DE NAVARRA

Estudiante
Marcos Rodrigo GALVEZ BELMONTE

Título
Sistema portable para localización de objetos y comunicación mediante audio 3D para personas invidentes

Grado
Grado en Ingeniería informática

Centro
Escuela superior de Ingeniería Industrial, Informática y Telecomunicación
Universidad Pública de Navarra

Director-a
Asier R. Marzo Pérez

Departamento
Estadística, Informática y Matemáticas

Curso académico
2019/2020

Semestre
Primavera

Resumen

En este trabajo se pretende desarrollar un sistema capaz de ayudar e indicar, mediante audio 3D, la localización de los objetos/obstáculos que tiene el usuario enfrente. Está orientado a ayudar a personas invidentes o con problemas de visión graves.

Para ello tendremos que utilizar técnicas de reconocimiento de imágenes por computador y Deep learning, para poder detectar dichos objetos e intentar reconocerlos para así poder nombrar al usuario el tipo de objeto que tenga delante.

Incluiré en este trabajo algo muy importante para su desarrollo, es una entrevista con una persona con problemas de visión grave, que me ayudará a entender la forma en la que se mueve por el mundo y cómo utiliza las distintas herramientas que existen hoy en día para poder sobrepasar los obstáculos que se le presentan.

Palabras clave: Audio 3d, reconocimiento de imágenes, Deep learning, personas invidentes, problemas de visión graves.

Abstract

The aim of this project is to make a system that can help and indicate through 3D Audio, the location of objects/obstacles which are in front of the user. It is focused on helping the blind or people whith major sight issues.

With that purpose, techniques of computer vision and Deep Learning have been used in order to detect those objects and try to recognize them so that the nature of the object can be informed to the user.

An interview with a person with major sight issues is included. This has helped me to understand the way he moves in the real world and how he uses the distincts tools that help him go through the obstacles every day.

Keywords: 3d Audio, computer vision, Deep learning, blind, people whith major sight issues.

Agradecimientos:

Quiero agradecer en especial a todas las personas que me han apoyado durante toda esta larga etapa de mi vida y me han animado a seguir adelante incluso en los momentos más difíciles en los que uno piensa en tirar la toalla. Sin ellas, este trabajo no estaría siquiera empezado.

Especial agradecimiento a:

Roberto Belmonte

Aurelia Janusis

Eliana Sánchez Violino

Alberto Modesto Galvez

Adriana Belmonte

Carlos Galvez

Asier Marzo Pérez

Sin orden alguno.

Índice

Introducción

A menudo podemos ver que las personas invidentes suelen ir de manera independiente por la calle sin ayuda de nadie, aunque muchas veces en que tienen que acudir a lugares cerrados como son centros comerciales, van acompañados de otras personas o perros lazarillos.

Esto hace pensar si realmente son suficientemente autónomas como para no depender de nadie a la hora de querer hacer una actividad que requiera guiarse entre obstáculos muy cercanos en lugares cerrados.

Como estudiante de ingeniería informática y conociendo todas las herramientas que día a día van saliendo a modo de tecnología y la velocidad con la que se van extendiendo, me resulta un poco preocupante ver como para muchas personas con dificultades aún no les ha llegado dicho avance.

Con motivo de mejorar la vida, libertad y autonomía de las personas con este tipo de discapacidad he decidido emplear los conocimientos obtenidos en mi carrera y utilizar el avance de las tecnologías para acercarlo a este campo.

Cabe destacar que existen varios proyectos que intentan mejorar la vida de las personas invidentes, Google (entre otras) es una empresa privada que está apostando mucho por ello. También existe un proyecto europeo (Sound Of Vision [1]) entre universidades con sede en Rumania que ha servido como ejemplo para este proyecto.

Objetivos

El objetivo de este trabajo final de grado es construir un modelo que ayude a las personas no videntes a llevar su día a día de forma más autónoma.

Para hacerlo, utilizaremos técnicas de visión por computador y audio 3D, además de distintas configuraciones de hardware para intentar hacer del modelo lo más portable y seguro posible.

Este trabajo está más orientado a la investigación que a construir un modelo óptimo final, por ello, iremos teniendo en cuenta las distintas configuraciones y distintas formas de comunicación con el usuario para intentar identificar cuál es la óptima a nivel de hardware y software.

El lenguaje de programación principal utilizado será Python y utilizaremos distintos equipos como fuente de procesamiento para poder encontrar cuál es el mejor en cuanto a tamaño y rendimiento. Para ello hemos seleccionado 4 dispositivos a los cuáles les haremos una serie de pruebas para comprobar rendimiento y portabilidad.

En cuanto al algoritmo que utilizaremos para la detección de objetos en la cámara será Yolo, el cual nos permitirá detectar todos los objetos en la imagen y así poder comunicarle de manera precisa a la persona que es lo que tiene en frente.

Es importante poder comunicarle al usuario desde distintos puntos de su rango auditivo, ya que de esta manera podremos detallarle además de la existencia de un objeto, la posición espacial del mismo. Por esto mismo es que necesitamos una librería de Python que nos permita manipular el audio para que dé un efecto de audio en tres dimensiones.

Para este tipo de pruebas, utilizaremos el software realizado para poder testear y entregar datos necesarios para la selección de los distintos hardware, como pueden ser, el ordenador (a utilizar), la cámara que utilizaremos para detectar los objetos, los instrumentos de medición de distancia y los auriculares utilizados para comunicar la información.

1. Fundamentación teórica

1.1. YOLO (You only look once)

YOLO es un algoritmo para el reconocimiento de objetos en imágenes en tiempo real desarrollado en la universidad de Cornell por Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick y Ali Farhadi.

A diferencia de los demás algoritmos, YOLO permite reconocer varios objetos haciendo sólo un barrido de la imagen y con sólo una red neuronal, haciéndolo más rápido que sus competidores.

A modo de curiosidad, la red neuronal base de YOLO puede funcionar a 45 fps en una GPU NVIDIA Titan X y una versión más rápida a 150 fps. Esto significa que se puede procesar video en streaming a tiempo real con menos de 25 milisegundos de latencia.

El funcionamiento es el siguiente:

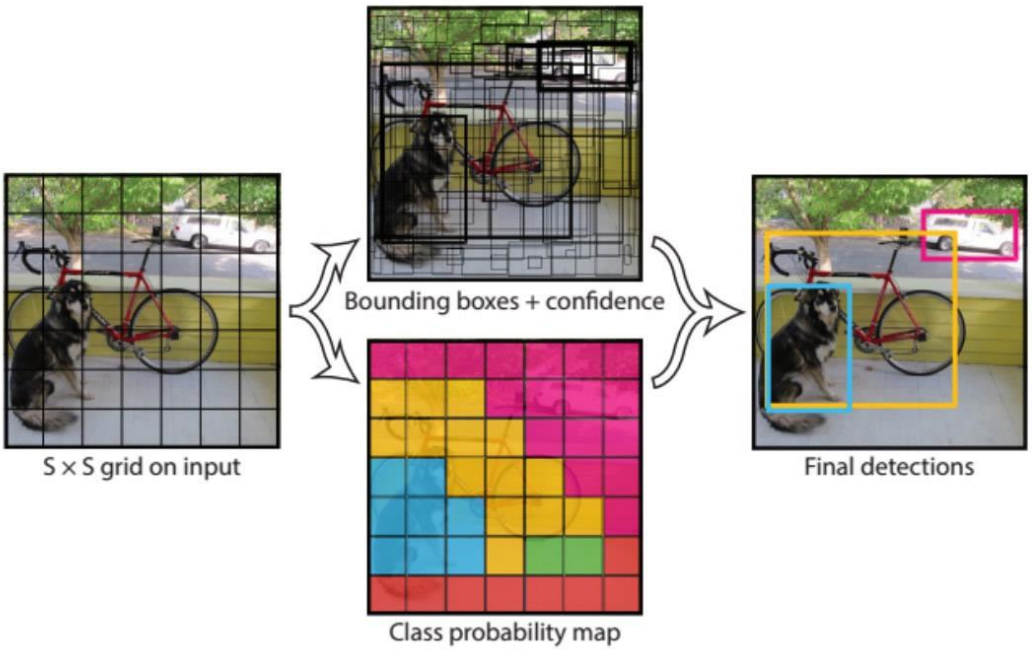

Figura 1: Explicación funcionamiento YOLO [2]

Este sistema de predicción detecta modelos como un problema de regresión. Divide la imagen en una matriz de tamaño S x S y para cada una de sus celdas predice B Bounding Boxes con la probabilidad para las C clases predichas. Estas predicciones son codificadas como un tensor S x S x (B*5+C).

Es decir, por cada subdivisión de la matriz, se califican las C clases con su respectiva probabilidad y luego al unificar todo, se unen las partes que pertenezcan a una clase en concreto, por ejemplo, en la Figura 1 el perro de abajo con su bounding box y su probabilidad.

La red neuronal convolucional utilizada por YOLO para la regresión tiene la siguiente forma:

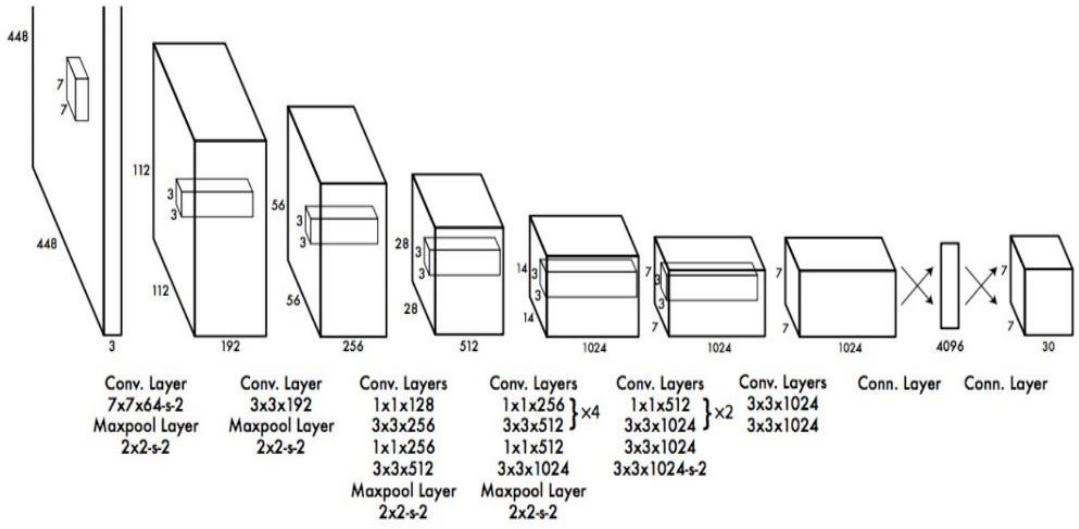

Figura 2: Imagen representativa de la red neuronal utilizada por YOLO [3]

Las capas convolucionales iniciales se encargan de obtener propiedades a partir de la imagen, mientras que las capas totalmente conectadas del final se encargan de predecir su probabilidad y coordenadas.

Esta red está basada en el modelo GoogLeNet para clasificación de imágenes. Está compuesta por 24 capas convolucionales seguida de dos capas totalmente conectadas.

La versión de TINY-YOLO utiliza 9 capas convolucionales en vez de 24 con menos filtros que la versión de 24, para hacerlo más rápido. Aunque ambas dos están entrenadas y testeadas con la misma cantidad de datos.

1.2. HRTF (Head-related transfer function)

El sonido antes de entrar en el conducto auditivo suele ser alterado por la posición desde donde surge el sonido y la posición real de la cabeza, torso y pabellón de la oreja.

La función HRTF se encarga de simular dichos cambios para que mediante auriculares podamos escuchar que el sonido viene de un sentido o de otro como si se tratase de la vida real.

Las distintas presiones de sonido binaural que resultan de la modificación por la física de nuestra cabeza, torso y oído, suele contener mucha información espacial la cual es utilizada por nuestro cerebro para determinar la localización de la fuente de sonido.

En la vida real el proceso de transmisión desde una fuente localizada de sonido suele representarse con un proceso lineal invariante en el tiempo (LTI) como se puede apreciar en la siguiente figura:

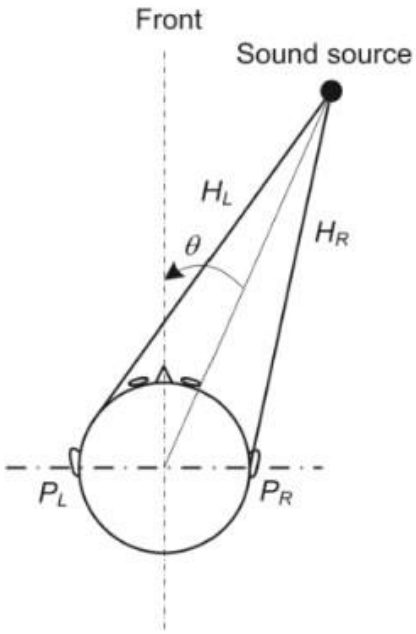

Figura 3: Transmisión de sonido desde un punto a cada uno de los oídos [4]

En la figura 3, podemos ver HL y HR que son la función HRTF que afecta en el oído izquierdo y derecho respectivamente, mientras que PL y PR representa la presión de sonido en el dominio de frecuencia tanto del oído izquierdo, como el derecho respectivamente. También podemos ver el ángulo que hay entre una línea recta trazada entre en centro de la cabeza y el origen del sonido y la línea recta que hay en el centro de la cabeza pasando por entre los ojos representado por la letra griega theta.

La transmisión de sonido entre una fuente de sonido y el tambor del oído se puede dividir en dos partes. La primera entre la fuente de sonido y la parte externa de la oreja, la cual es dependiente de la dirección de donde proviene el sonido y posición de la cabeza. La segunda entre el canal auditivo y el tambor del oído, la cual es independiente de la dirección.

La siguiente figura, nos muestra cómo se subdividen estas dos etapas:

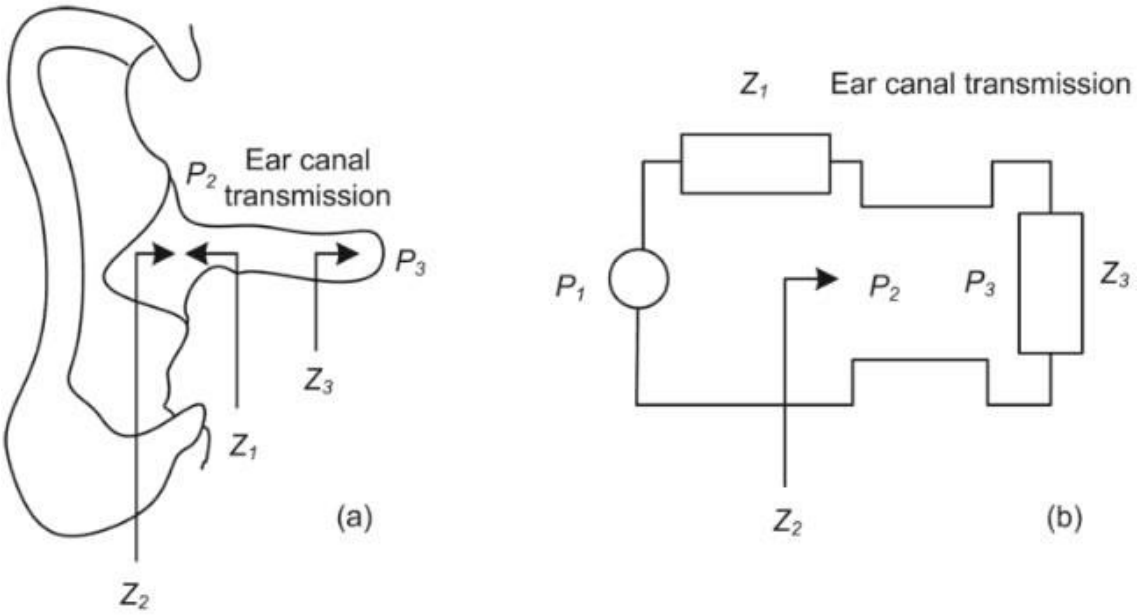

Figura 4: Separación de la transmisión de sonido direccional en dos etapas [5]

En la imagen se pueden apreciar las presiones dentro del canal auditivo P1, P2 y P3 junto con Z1, Z2 y Z3. Donde Z1 es la radiación de impedancia a la entrada de canal auditivo desde dentro, Z2 es la radiación de impedancia desde la parte externa del canal auditivo hacia adentro y Z3 es la impedancia del tambor.

El audio 3d o audio espacial es uno de los métodos más utilizados en videojuegos o sistemas de entretenimiento, ya que, tanto de manera física como de manera virtual, se intenta que la persona que está disfrutando dicho entretenimiento, ya sea una película, un videojuego, etc, pueda sentirse dentro de la misma dándole así un nivel de realismo que el espectador quiere sentir.

1.3. Entrevista

En este encuentro he entrevistado a un hombre de 43 años con un grado de discapacidad del 77%. Tengo que aclarar que para transcribir la entrevista he estado investigando sobre su problema de visión para poder entenderlo más a fondo. Hemos estado dialogando relajadamente durante dos horas y veinte minutos, conversación que después he vuelto a escuchar, disfrutar y examinar. Es de agradecer haber podido hablar con personas tan abiertas que nos cuentan una parte del mundo y de sus vidas que en principio no conocemos con tanta profundidad.

Marcos:

- La primera pregunta, ¿cuál es realmente tu problema? ¿Tiene algún nombre en especial?

Entrevistado:

- Bueno, problemas tenemos todos, muchos o pocos, pero el problema que tiene que ver con el tema de lo visual, en realidad hay un porcentaje altísimo de la población mundial que no ve bien. Claro, pero pasa que las maneras de no ver bien son, primero muy diversas y luego hay grados, además hay gente que va involucionando, por ejemplo, hay gente que ha visto mal siempre o hay gente que ha visto mal a partir de una edad por degeneración del ojo, o gente que puede tener un accidente y su visión se pierde o deteriora.
- Yo siempre he visto mal, cuando era mucho más joven sí que veía mejor. Yo en principio tenía diagnosticado miopía y astigmatismo. Lo que sabía es que no era una miopía normal porque conocía mucha gente que tenía miopía y se ponían gafas y no veían perfecto, pero si corregían muchísimo su visión, lo que a mí me pasaba es que mis problemas mayores, con las gafas no mejoraban mucho. Entonces ya ahí intuía yo que tenía algo diferente.
- A partir de 2003 aproximadamente descubrí que lo que tenía yo era una enfermedad que se llama retinosis pigmentaria, que es un problema de la retina que se pigmenta, en vez de ser transparente tiene una opacidad.
- Dentro de esa enfermedad como en todas las demás también hay muchos grados, hay personas que están completamente ciegas y luego hay gente que les afecta, pero muy poquito. Yo estoy un poco entre medias, todavía tengo resto visual, pero tengo una serie de problemas visuales importantes.

Marcos:

- Lo que sería la siguiente pregunta en base a lo que acabas de contestar es, ¿puedes describirme cómo sería ver a través de tus ojos?

Entrevistado:

- Hay una parte objetiva y una parte subjetiva, la parte objetiva es cuando un médico mira el ojo con sus instrumentos, y él mismo observa que el ojo tiene un problema en su fisionomía.
- La parte subjetiva es la que el médico no puede conocer bien porque él sabe que hay algo que está mal pero no sabe cómo afecta exactamente eso a mi visión por el cerebro.

Hay varias cosas, lo principal, por ejemplo, estas en un baño y te duchas y tienes ahí como una niebla que tú, después de ducharte, si miras al espejo ves que el espejo está empañado y tienes una muy mala visión, que generalmente lo corriges pasando una toalla o un pañuelo por el espejo. Normalmente cuando miras al espejo no puedes decir que no ves nada, sí que ves, pero ves mal. Esa es la sensación que tengo yo normalmente. No es que las gafas estén empañadas, sino que lo que tiene esa opacidad es mi ojo, es un problema de mi retina.

- Otro caso es cuando pongo un folio blanco enfrente mío, que yo sé que es nuevo y por tanto completamente blanco, pero lo veo como si estuviera sucio, entonces sé que soy yo que veo mal ya que el papel es nuevo, lo acabo de sacar de su paquete y sé que es objetivamente blanco que no está sucio. El problema no está en ese papel sino en mi visión, en mi discapacidad para verlo como realmente es, blanco y sin manchas.

Marcos:

- Entonces, ¿ves mal tanto de lejos como de cerca?

Entrevistado:

- Claro, porque, si el espejo está empañado, vas a ver mal siempre, este el espejo cerca o lejos.
- Otro problema grande que padezco es el problema con la luz, esto me lo han explicado los médicos, la luz en vez de entrar de una forma directa al ojo y de ahí al nervio óptico y luego al cerebro, al tener esa opacidad en los líquidos del ojo, la luz en vez de ir por un camino recto empieza a rebotar ya que encuentra resistencias.
- Entonces, esto me genera que cuando hay mucha luz es como si me deslumbrasen con una linterna en los ojos. Esto durante el día lo corrijo con unas gafas de sol con unos filtros un poco especiales. Cuando hay poca luz veo muy mal, veo muchísimo menos que cualquier persona, veo mucho más oscuro que cualquier otra persona con buena salud visual.
- Esos son los principales problemas que tengo yo, hay otra gente que tiene también retinosis pigmentaria que tiene otros problemas, o los mismos, pero más acrecentados. Por ejemplo, otro problema grande es que hay muchos que tienen una gran reducción de su campo visual. Yo ese problema no lo noto mucho, sé por estudios médicos que me han hecho que tengo un campo visual muy reducido, de sobre un 10%, pero yo eso no lo noto así. Será que como nunca he visto bien tampoco puedo comparar y saber qué significa en realdad ver bien y cómo veo de mal.
- En ese sentido la informática ha avanzado muchísimo ya que existen programas que calculan este tipo de cosas, entonces una persona que ve bien va haciendo click en una pantalla de móvil o de ordenador y seleccionando el tipo de discapacidades visuales y el programa le da una aproximación de cómo vería esa persona con cada una de esas discapacidades.

Marcos:

- ¿Existe algún tipo de cura hoy en día, tratamiento o cirugía?

Entrevistado:

- Hoy en día no existe tratamiento ni cirugía para corregir esto. Por eso mucha gente intenta cuidarse y revisa cómo come, sus hábitos de vida, para que esto no le vaya a más, no le empeore mucho o rápido.

Marcos:

- Tras comentar el proyecto con el entrevistado, le pregunté, ¿qué te parece mi proyecto?

Entrevistado:

- Yo creo que en lo que estás trabajando va orientado a una persona que ve muy mal, seguramente ve peor que yo. Seguramente una persona que tiene retinosis pigmentaria, pero que, si ve mejor que yo, no le va a aportar mucho ese sistema que estás diseñando.
- Hay personas que son casi ciegas, o que no ven nada. Entonces para gente así lo que tú le das es una aportación enorme, porque van a sentirse mucho más autónomos. Estas personas por ejemplo se apoyan muchísimo en el oído y también se apoyan muchísimo en su bastón. Entonces lo que tú les vas a poner no van a hacer que dejen su bastón en casa, sino que va a ser un complemento que les dé más autonomía, pero quizá siempre como un buen complemento.
- Tu proyecto quizá no esté orientado para gente que ve mal, sino para gente que ve mal, pero con un grado de discapacidad grande o muy grande.
- Estaría bien que se le indique cosas importantes que pueden llegar a hacerles daño o tropezar con mayor importancia que otras que son secundarias. Por ejemplo, es mejor que le avise que hay una bici de un niño tirada en el suelo antes de que se avise de que hay una moneda de 5 céntimos.
- Por ejemplo, nosotros, al no ver bien, casi siempre nos movemos por los mismos sitios, ya que vamos memorizando el recorrido y nos sentimos más seguros yendo por ahí, pero de repente un día ponen una valla de construcción en el suelo y no la vemos y nos podemos caer o golpear, entonces esas cosas son las que importan más a la hora de indicar cuando caminamos por la calle, las cosas que no nos esperamos que estén ahí, por ejemplo en mitad de una acera.
- Cuando salimos a la calle tenemos más interés en no tropezarnos y no lastimarnos que en saber lo que hay realmente, a veces es más importante decir que hay algo con lo que te puedes llegar a chocar que decir realmente lo que es. Por ejemplo, puede pasar que un andamio sea difícil de reconocer por un sistema informático lo que es, pero que puedes llegar a encontrarte con él y que se te diga “objeto peligroso a 3 metros”, antes que nombrar que hay un coche, porque entonces tú ya te paras y andas con mucho más cuidado, tanteas con el bastón o pides ayuda a alguien.

Marcos:

- Perdón que te interrumpa, ahora que acabas de nombrar la distancia hacia el objeto, ¿es mejor que diga la distancia o que lo diga más cerca o más lejos?

Entrevistado:

- Yo creo que sí, que sería mejor indicar la distancia. Yo creo que el perfil del usuario al que estás orientando esa ayuda es una persona que ve muy mal, y va a andar despacio. Así que yo creo que lo mejor sería decir la distancia a la que están las cosas, esos posibles

peligros u obstáculos. No es necesario decir exactamente la distancia, puede ser aproximado, por ejemplo, está a 3 metros, 2 metros, 1 metros, o menos de 1 metro.
- La persona no está buscando una precisión elevada, sino que le ponga en alerta de que ahí hay algo, un posible peligro u obstáculo.

Marcos:

- ¿Qué información se obtiene del bastón?

Entrevistado:

- Del bastón tienes muchísima información. Incluso es información para los demás, porque cuando tú ves a alguien con bastón ya sabes que ve mal. Te puede pasar que al ir por la calle sin el bastón te puedes llegar a chocar con una persona y esta puede pensar o que estás borracho o drogado o hasta que le quiero robar, ya que a veces no saben que ves mal, porque a veces una persona que ve mal no lleva unas gafas con cristales muy gruesos, como ocurría hace unos años, sin embargo, llevas el bastón y ya saben que no ves bien, no hay malentendidos y tienen más cuidado. El bastón también hace ruido, entonces si la persona está despistada, o escuchando música o mirando el móvil, el ruido del bastón le llama la atención, te ven y evitan chocarse.
- El bastón te da muchísima información. Yo a veces me fío mucho más del bastón que de las personas, porque muchas veces las personas, aunque quieren ayudarte, hay muchas que no están acostumbradas a guiar a alguien y se les olvida decirte que había un escalón, o te dicen sólo “cuidado” y tú no sabes si es una rama de un árbol, si es una bicicleta o un escalón y te preguntas, ¿cuidado con qué? En cambio, con el bastón sabes si es un escalón, sabes si es para arriba o para abajo, sabes su profundidad o su altura, si hay más escalones porque es una escalera e incluso izquierda y derecha. Hay mucha gente que me quiere ayudar y se lía al decirme si es izquierda o derecha. En cambio, el bastón no, el bastón siempre que detecta algo te da todos los detalles y con 100% de rapidez y fiabilidad. El bastón nunca se va a equivocar, aunque pueden existir algunas cosas que no detecte, como una rama de un árbol, por ejemplo.
- Yo antes de tener bastón no iba a muchos sitios por evitar problemas. Pero con el bastón siento esa seguridad que me permite ir a tomar un café o ir a comer solo a un restaurante.
- EL bastón nos enseñan a manejarlo, la gente piensa que ves mal, te dan el bastón y ya está. No es como conducir un coche, pero sí que te dan unas clases que te van enseñando trucos de cómo tienes que moverlo, hay diferentes tipos de bastones. No es lo mismo andar por la calle que hacerlo en un establecimiento interior. Entonces yo creo que tu trabajo es un complemento para el bastón y no hay que pensarlos en separado. El bastón es eficiente, barato y no casi nunca se avería, y las personas que ven mal lo seguirían utilizando, sobre todo en la calle.
- A mí el bastón me ayuda muchísimo a localizar objetos que hay en el suelo, pero muchos de los accidentes que yo he tenido es con objetos que no están en el suelo, he tenido ya alguno un poco grave y es por ejemplo con las cosas que están elevadas del suelo, que el bastón nunca me va a poder detectar. Un ejemplo es cuando una persona pone el coche girado hacia atrás para descargar cosas y deja la puerta del maletero abierta, eso queda a la altura de la frente y es un peligro muy grande que el bastón no lo detecta. Eso sí que es principal que el dispositivo te lo anuncie para que no sucedan este tipo de accidentes. Por eso digo que lo más importante cuando vas por la calle es que te evite

el choque a que te diga exactamente lo que tienes adelante, por si impidiera tu trayecto y pudiera generarte un accidente.
- En el ámbito doméstico es otra cosa, ya que en interiores ya conoces las cosas que te puedas llegar a encontrar, y con el bastón puedes detectar las cosas con las que puedes chocarte. Ahí sí que es importante que te diga por ejemplo, “en esta mesa tienes el el móvil, las llaves, la cartera” o “en el suelo está un billete o el cargador del móvil”, por si se te ha caído algo y puedes llegar a perderlo.
- Esa posible lectura de objetos, que te indique lo que es y dónde está, está bien para lugares internos, el ámbito doméstico. Cuando estás en la calle lo que más nos preocupa es la colisión, ir por un sitio peatonal y chocarnos con algo peligroso e inesperado.

Marcos:

- Otras de las cosas que probamos en mi proyecto es el dispositivo de sonido para comunicar al usuario los objetos, estamos entre los auriculares normales internos y los que son de conducción ósea.
- En particular mis preferidos son los de conducción ósea porque no le quitan la audición a la persona usuaria.

Entrevistado:

- No es que te vengan superpoderes con un sentido cuando pierdes otro. Pero en mi caso, como tengo la visión limitada, afinas o aprovechas más otros sentidos como la audición. Acabas agudizando y estando mucho más atento a la audición porque centras más tu atención en ello.
- Por eso lo que tú dices es muy importante, ya que si tienes una cosa interna con un sistema informático que te transmite muy bien la información, pero te priva del resto de información auditiva que tienes del exterior. Incluso hay gente que tiene en un oído un casco y el otro libre, ya que no quieres perderte toda esa información tan importante que te viene del exterior, que te avisa de una bici que pasa al lado, un coche que viene por detrás, una persona con un niño que caminan por delante, unos albañiles que trabajan en una obra a tu izquierda, o lo que sea.

Marcos:

- Me has ayudado mucho a darme cuenta mejor de los distintos enfoques que no había tenido muy en cuenta, por ejemplo, esto de enfocar dos situaciones distintas cuando uno está en la calle que cuando uno está dentro de un establecimiento o en casa. Si bien las he tenido en cuenta siempre, no sabía muy bien cuál sería la diferencia exacta para tener esto en cuenta y me lo has hecho ver con más claridad.
- Por otro lado, cambiar un poco el pensamiento de indicar absolutamente todo lo que se ve o decirle por dónde tiene que ir pensando en que esa persona ya no necesita el bastón, para tener en cuenta el bastón y complementarlo con mi dispositivo para mejorar realmente la vida de esa persona y darle más información, seguridad y, en definitiva, autonomía.
- ¡Gracias por tu colaboración!

Entrevistado:

- Gracias a ti, Marcos.
- ¡Ha sido un placer!

2. Desarrollo Práctico

2.1. Fases del proyecto

El proyecto se divide en tres etapas claras, que son:

1. Recolección de datos.
2. Procesamiento de datos.
3. Comunicación con el usuario.

Figura 5: Las tres etapas del proyecto

2.2. Fase de obtención de datos

Mediante reconocimiento de imagen vamos a detectar posibles obstáculos para la persona.

Por otro lado, vamos a necesitar obtener la distancia entre la persona y cada uno de los objetos.

Por eso mismo vamos a dividir la etapa de recolección de datos en dos sub-etapas llamadas:

1. Reconocimiento de imágenes.
2. Medición de distancia.

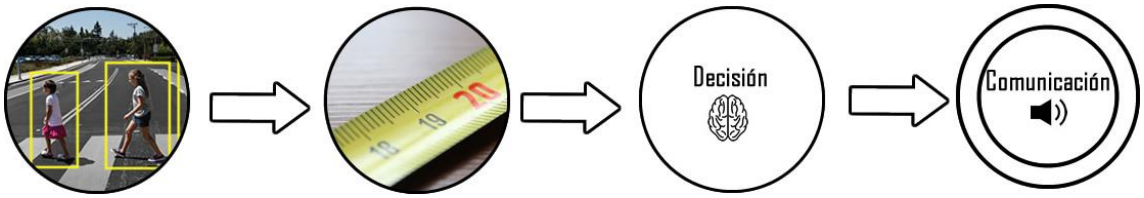

Figura 6: Secuencia principal de funcionamiento del sistema.

Para cada sub-etapa vamos a testear los diferentes hardware con los que contamos para poder tomar una decisión sobre cuál es el mejor para luego poder presentar el prototipo final.

Hay que tener en cuenta que la parte de tomar decisiones sobre los datos que tenemos es pura y exclusivamente software.

2.2.1. Hardware para el reconocimiento de objetos en imágenes

Para poder hacer una elección razonable vamos a tener que analizar:

1. La cámara que vamos a utilizar.
2. El modelo de YOLO a utilizar

El algoritmo de reconocimiento de imágenes que utilizaremos será YOLO.

YOLO (You only look once – Sólo miras una vez) como su nombre lo indica, nos permite obtener una imagen clasificada sólo haciendo un barrido completo de la imagen.

A diferencia de otros algoritmos que tienen que recorrer la imagen varias veces, YOLO permite mejorar los tiempos y obtener un rendimiento igual o superior al de sus competidores.

Trabajaremos con un framework llamado Darkflow, el cuál es una traducción a Tensorflow del original Darknet que está escrito en C++.

Como modelo de datos para obtener los distintos objetos utilizaremos una librería que viene con Darkflow llamada cocos. La cuál es una librería de clases para detección de objetos hecha por Google.

Las versiones Tiny de YOLO, nos ofrece una red neuronal más ligera y con menos capas que las Yolo normales, aunque empeora un poco la exactitud de la clasificación. Lo ideal será utilizar la versión completa de YOLO, aunque será un poco difícil obtener un buen rendimiento con los ordenadores más portables.

Para configurar YOLO necesitamos utilizar la librería OpenCV de Python con Tensorflow para los ordenadores con tarjetas gráficas Nvidia. Cabe destacar que para reconocimiento de imágenes lo más importante es la tarjeta gráfica y la memoria RAM que tengamos, ya que las gpus son más potentes a nivel gráfico que los cpu. A priori podemos pensar que entre todos los ordenadores que probemos, los que tengan tarjeta gráfica serán mejores que los que no. Aunque también cuánto mejor tarjeta gráfica tenga, más potencia consumirán y mayor será la batería necesaria para lograr el mayor rendimiento.

A continuación, se presentan tres tablas; en la tabla 1 se puede observar una comparación entre los modelos de YOLO y TINY YOLO donde se señalan los pro y contras de cada uno.

En la tabla 2 se muestran las particularidades de las cinco posibles cámaras a utilizar y por último en la tabla 3, se detallan las características de los cuatro posibles ordenadores a utilizar.

Modelo	Pros	Contras
YOLO	- Buena precisión.	- Muy costoso para la Gpu ya que tiene muchas capas.
TINY-YOLO	- Bastante ligera para la Gpu ya que tiene pocas capas.	- Más imprecisa.

Tabla 1: Comparación entre modelos de YOLO.

Hardware	Pros	Contras	Tamaño	Precio
Cámara raspi V2	- Buena latencia. - Reconoce bien los datos.	- No tiene buena resolución, pero si la justa para lo que necesitamos.	2,5 x 2,4 x 0,9 cm	29,46€
Webcam full hd	- Buena definición.	- Mucha latencia en ordenadores pequeños.	12 x 9 x 6,6 cm	~22€
Cámara raspi gran angular	- Definición aceptable. - Mas ángulo de visión.	- Mas latencia que la cámara v2. - Deforma un poco la imagen al ser gran angular.	11,8 x 5,2 x 2,8 cm	23,98€
Kinect cámara	- Buena latencia. - Cámara Depth integrada.	- No es muy portable.	24,9 x 6,6 x 6,7 cm	~50€
Intel Realsense	- Cámara Depth int. - Mejor calidad imag. - Portable.	- Alto precio.	9 x 2,5 x 2,5 cm	205€

Tabla 2: Comparación entre posibles cámaras a utilizar.

Otro aspecto fundamental para la obtención de datos a partir de las imágenes obtenidas es el ordenador que utilicemos.

Estamos buscando algo que sea portable y que nos permita utilizar YOLO sin que los FPS sean demasiado.

De esta manera he creado una tabla comparativa de los distintos ordenadores que disponemos para poder evaluar su rendimiento y portabilidad.

Hardware	CPU (Frec. Max. GHz)	Núcleos (CPU)	GPU (Cores)	Ram (GPU)	Ram (Gb)	Consumo	Peso	Precio
Raspberry Pi 3 b+	1,4	4	---------	Comp.	1	6W	86,2 g	39,91€
Raspberry Pi 4	1,5	4	---------	Comp.	4	8W	59g	64,99€
NVIDIA Jetson Nano	2	4	128 Cuda	Comp.	4	20W	249g	162,90 €
Ordenador portátil	4	4	640 Cuda	4	16	130W	2,3K g	899€

Tabla 3: Comparación entre los posibles ordenadores a utilizar.

Pruebas realizadas:

En estas pruebas, vamos a utilizar las distintas configuraciones de YOLO y TINY-YOLO con los distintos ordenadores y cámaras para intentar obtener el mejor rendimiento en fps y que el ordenador utilizado sea lo más portable posible.

Para cada prueba vamos a dejar funcionando la cámara durante 1 minuto y obtendremos los valores máximo y mínimo de fps para calcular la media aritmética. De esta manera obtendremos un valor medio para la columna de fps.

1. Raspberry Pi 3 b+:

Con este ordenador no he logrado que ejecute ningún tipo de YOLO ya que la memoria RAM es muy poca para ejecutarlos, por ende, no tengo capturas de los resultados de los experimentos.

2. Raspberry Pi 4 b:

- TINY-YOLO:

La primera cámara a analizar es la V2 para raspberry pi.

Como se puede observar en la siguiente imagen, los resultados son parecidos:

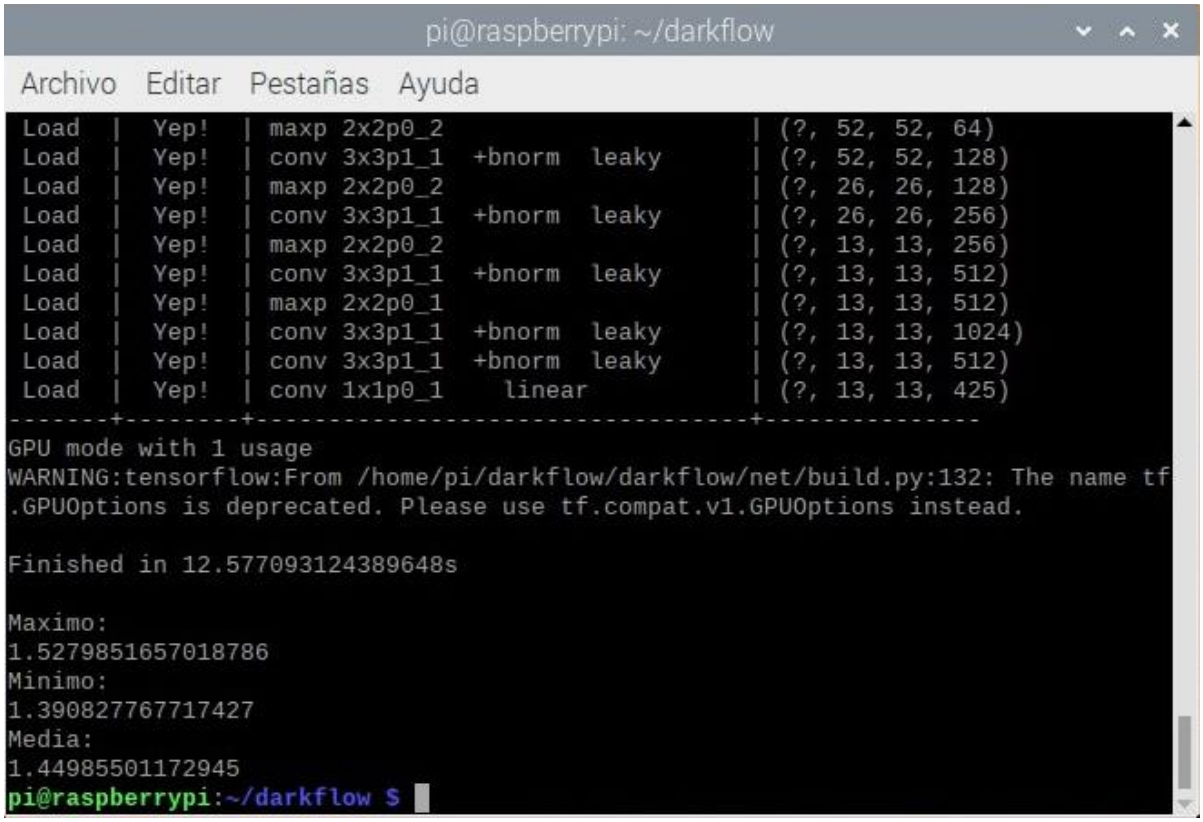

Figura 7: Resultados de pruebas realizadas con TINY-YOLO en Raspberry Pi 4 b con la cámara V2.

La siguiente cámara que probaremos será la Rev 2.2 gran angular, que presento en la página siguiente:

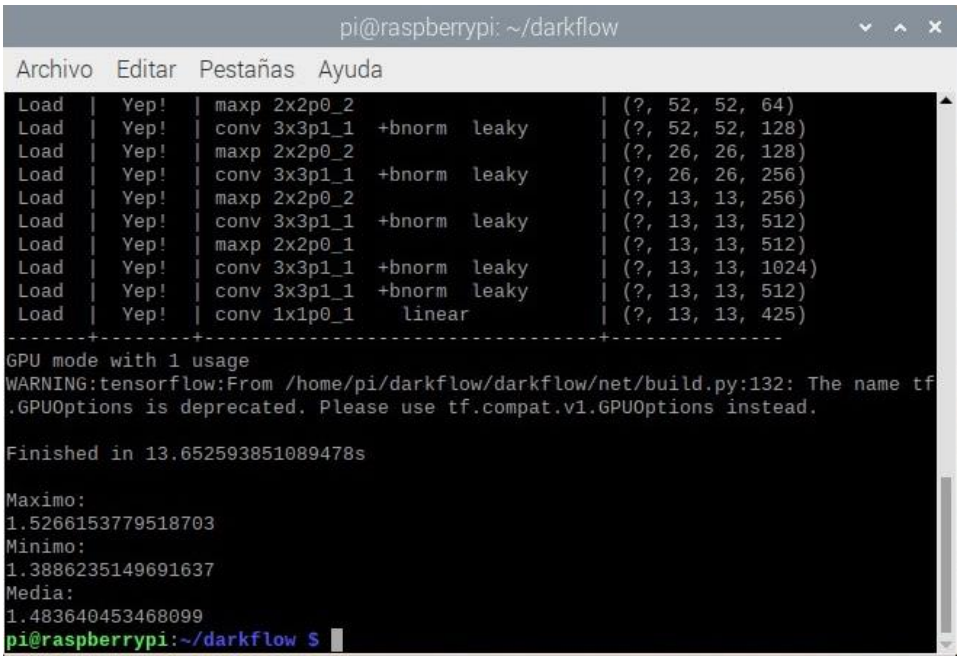

Figura 8: Resultados de pruebas realizadas con TINY-YOLO en Raspberry Pi 4 b con la cámara Rev 2.2 gran angular.

A continuación, probamos la webcam Full HD y los resultados fueron los siguientes:

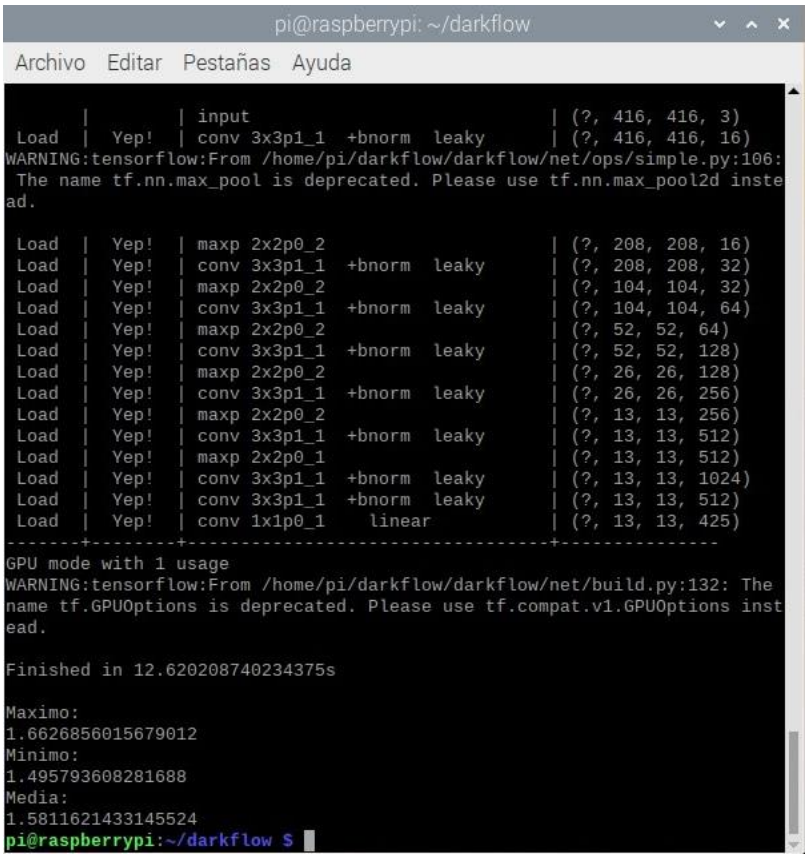

Figura 9: Resultados de pruebas realizadas con TINY-YOLO en Raspberry Pi 4 b con la webcam FHD.

Vale acotar en este caso que la imagen tiene dentro de todo unos fps razonables, pero lo que va muy lento es la latencia, ya que la imagen se ve fluida pero los movimientos se captan unos milisegundos después de haberlos hecho.

Por último, probaremos con la Kinect, obteniendo los siguientes resultados:

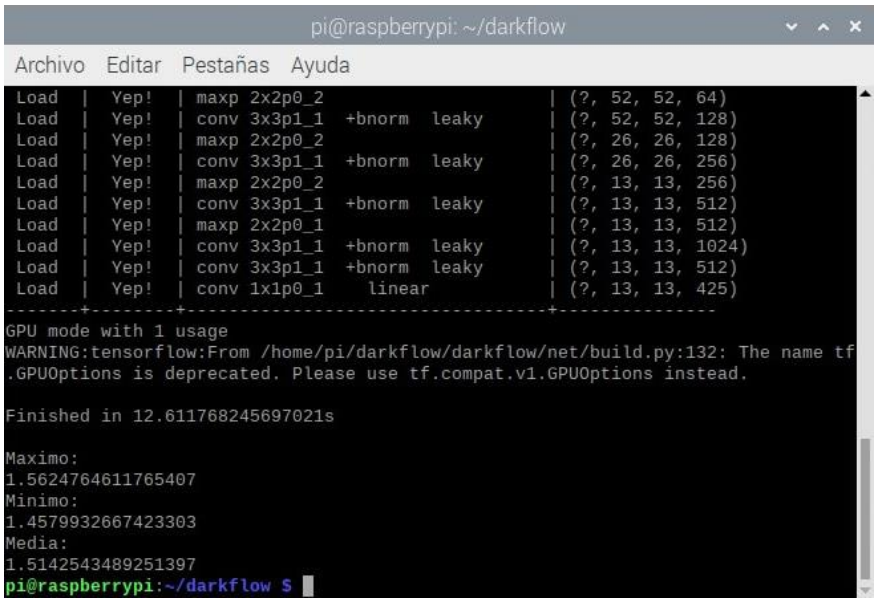

Figura 10: Resultados de pruebas realizadas con TINY-YOLO en Raspberry Pi 4 b con la cámara Kinect.

- YOLO:

El proceso de ejecución se interrumpe con todas las cámaras por falta de memoria.

3. Nvidia Jetson Nano:

- TINY-YOLO:

Primero analizaremos con la cámara de Raspberry pi V2. Como se puede observar en la siguiente imagen, los resultados fueron parecidos:

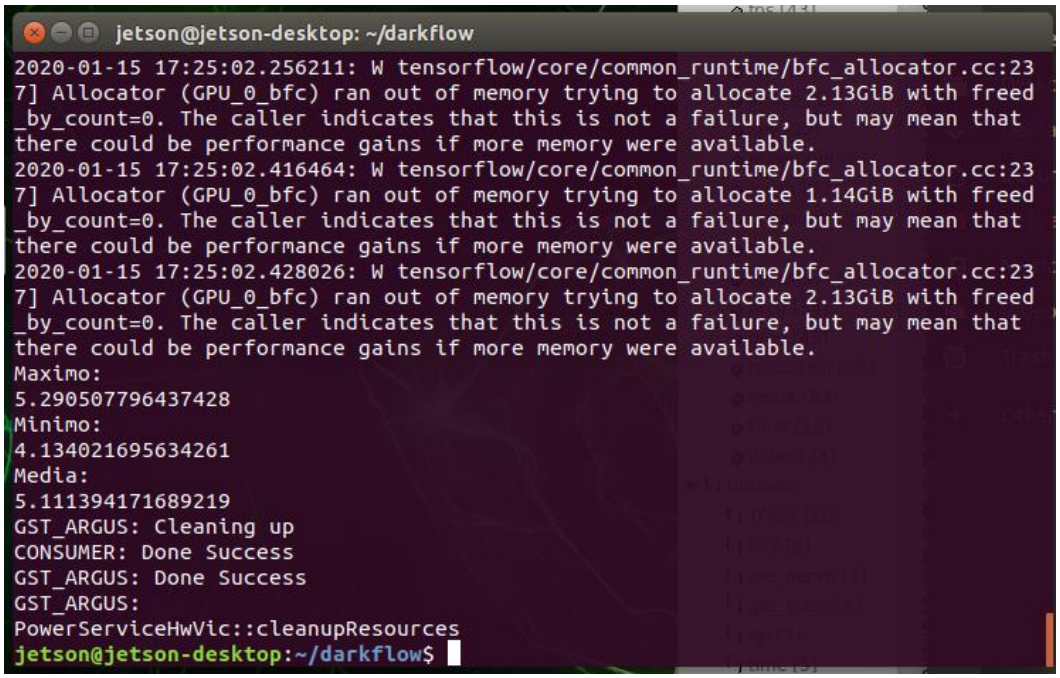

Figura 11: Resultados de pruebas realizadas con TINY-YOLO en NVIDIA Jetson Nano con la cámara V2.

Nota: La configuración necesaria para que la imagen de la cámara se muestre y no se vea una imagen verde, hace que YOLO tenga problemas para detectar o identificar objetos.

La siguiente cámara que probaremos será la Rev 2.2 gran angular. En este caso resultó imposible debido a que la Jetson Nano no soporta esta cámara.

A continuación, probamos la webcam Full HD y los resultados fueron los siguientes:

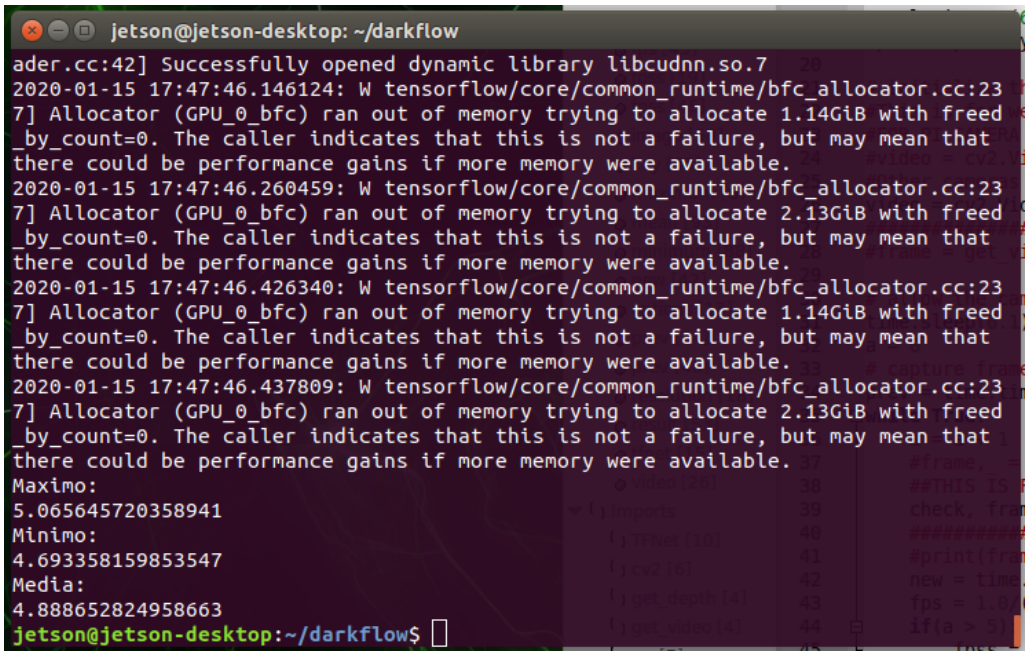

Figura 12: Resultados de pruebas realizadas con TINY-YOLO en NVIDIA Jetson Nano con la webcam FHD.

Por último, probamos con la Kinect con los siguientes resultados:

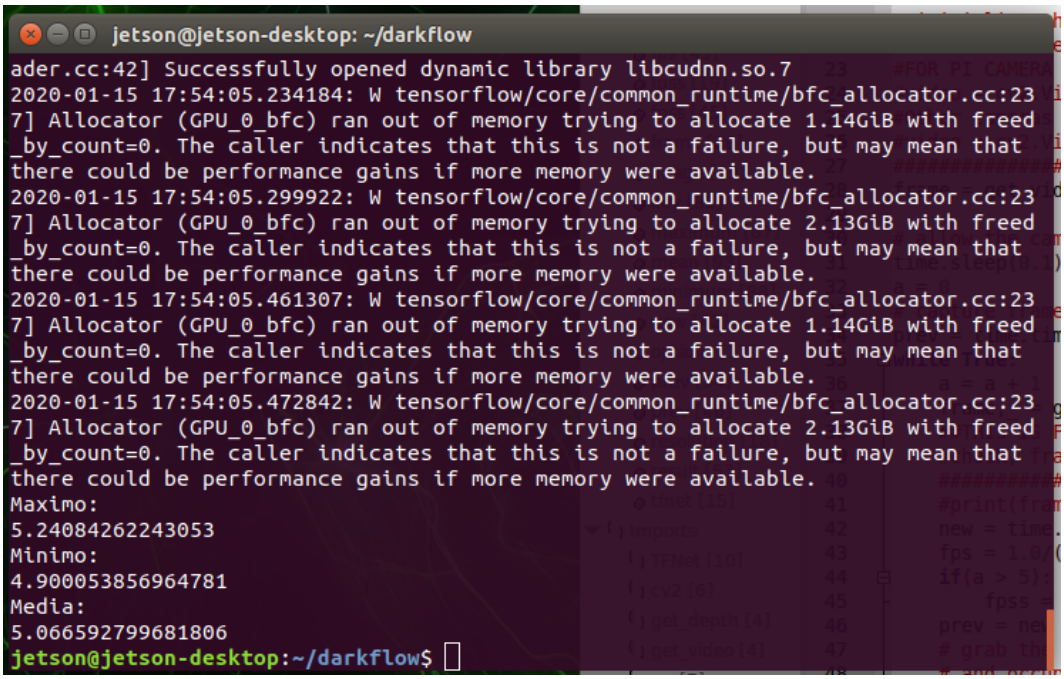

Figura 13: Resultados de pruebas realizadas con TINY-YOLO en NVIDIA Jetson Nano con la cámara Kinect.

- YOLO:

El proceso de ejecución se interrumpe con todas las cámaras por falta de memoria.

4. Ordenador portátil:

Para el ordenador sólo vamos a poder probar dos cámaras, la webcam FHD y la Kinect.

- TINY-YOLO:

Empezamos con la webcam FHD y obtenemos los siguientes datos:

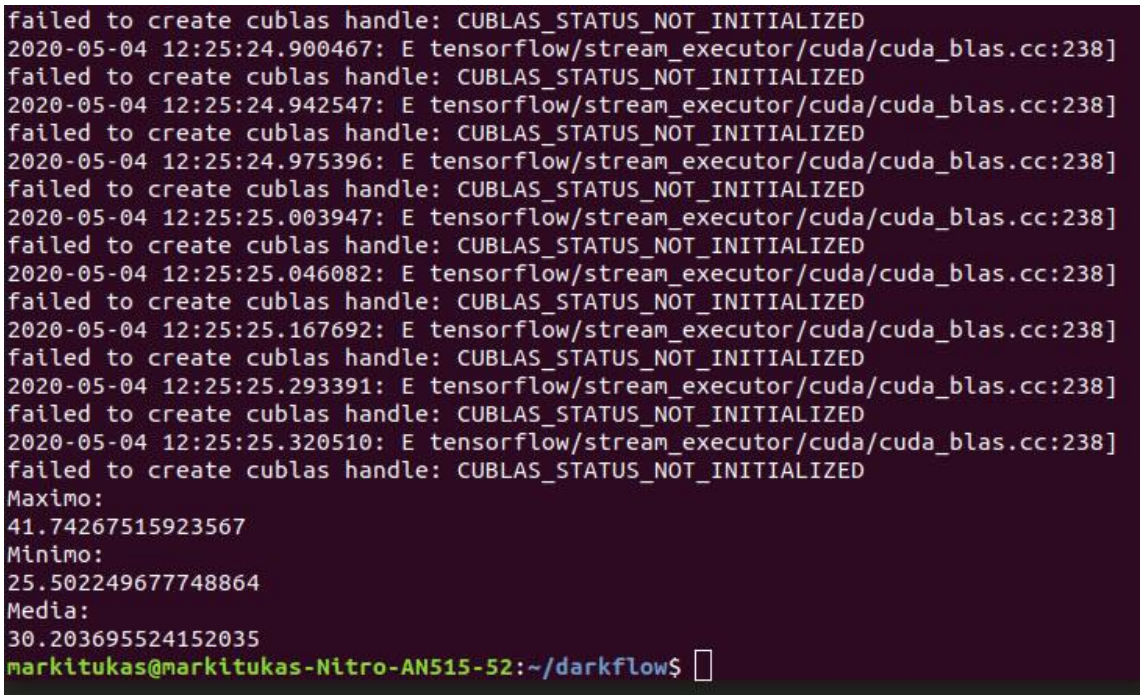

Figura 14: Resultados de pruebas realizadas con TINY-YOLO en Ordenador portátil con la webcam FHD.

A continuación, probamos con la Kinect y los resultados fueron muy buenos:

Figura 15: Resultados de pruebas realizadas con TINY-YOLO en Ordenador portátil con la cámara Kinect.

- YOLO:

Sólo se ejecuta en pc, así que comenzaremos con la webcam FHD:

```
WARNING:tensorflow:From /home/markitukas/darkflow/darkflow/net/build.py:149: The
 name tf.train.Saver is deprecated. Please use tf.compat.v1.train.Saver instead.

WARNING:tensorflow:From /home/markitukas/darkflow/darkflow/net/build.py:149: The
 name tf.global_variables is deprecated. Please use tf.compat.v1.global_variable
s instead.

Finished in 3.1963248252868652s

2020-05-04 12:22:51.097702: I tensorflow/stream_executor/platform/default/dso_lo
ader.cc:44] Successfully opened dynamic library libcudnn.so.7
2020-05-04 12:22:51.875930: I tensorflow/stream_executor/platform/default/dso_lo
ader.cc:44] Successfully opened dynamic library libcublas.so.10.0
2020-05-04 12:22:52.697231: W tensorflow/core/common_runtime/bfc_allocator.cc:23
9] Allocator (GPU_0_bfc) ran out of memory trying to allocate 3.55GiB with freed
_by_count=0. The caller indicates that this is not a failure, but may mean that
there could be performance gains if more memory were available.
Maximo:
10.602868684621646
Minimo:
9.520629941277814
Media:
10.08075001677002
markitukas@markitukas-Nitro-AN515-52:~/darkflow$
```

Figura 16: Resultados de pruebas realizadas con YOLO en Ordenador portátil con la webcam FHD.

Y por último vamos a probar YOLO con la Kinect:

```
9] Allocator (GPU_0_bfc) ran out of memory trying to allocate 1.75GiB with freed
_by_count=0. The caller indicates that this is not a failure, but may mean that
there could be performance gains if more memory were available.
2020-05-04 12:39:15.637172: I tensorflow/stream_executor/platform/default/dso_lo
ader.cc:44] Successfully opened dynamic library libcublas.so.10.0
2020-05-04 12:39:15.819828: W tensorflow/core/common_runtime/bfc_allocator.cc:23
9] Allocator (GPU_0_bfc) ran out of memory trying to allocate 1.21GiB with freed
_by_count=0. The caller indicates that this is not a failure, but may mean that
there could be performance gains if more memory were available.
2020-05-04 12:39:16.142468: W tensorflow/core/common_runtime/bfc_allocator.cc:23
9] Allocator (GPU_0_bfc) ran out of memory trying to allocate 2.15GiB with freed
_by_count=0. The caller indicates that this is not a failure, but may mean that
there could be performance gains if more memory were available.
2020-05-04 12:39:16.358343: W tensorflow/core/common_runtime/bfc_allocator.cc:23
9] Allocator (GPU_0_bfc) ran out of memory trying to allocate 3.55GiB with freed
_by_count=0. The caller indicates that this is not a failure, but may mean that
there could be performance gains if more memory were available.
Maximo:
11.0557386037145
Minimo:
9.753673640712242
Media:
10.375666539808124
markitukas@markitukas-Nitro-AN515-52:~/darkflow$
```

Figura 17: Resultados de pruebas realizadas con YOLO en Ordenador portátil con la cámara Kinect.

Ordenador	Cámara	Fps medios YOLO V2	Fps medios TINY-YOLO V2
Raspberry Pi 3 b+	V2	No Ejecuta	No Ejecuta
Raspberry Pi 3 b+	Rev 2.2 gran angular	No Ejecuta	No Ejecuta
Raspberry Pi 3 b+	Webcam fhd	No Ejecuta	No Ejecuta
Raspberry Pi 3 b+	Kinect	No Ejecuta	No Ejecuta
Raspberry Pi 4	V2	No Ejecuta	1.45
Raspberry Pi 4	Rev 2.2 gran angular	No Ejecuta	1.48
Raspberry Pi 4	Webcam fhd	No Ejecuta	1.5(retardo)
Raspberry Pi 4	Kinect	No Ejecuta	1.46
NVIDIA Jetson Nano	V2	No Ejecuta	5.11
NVIDIA Jetson Nano	Webcam fhd	No Ejecuta	4.89
NVIDIA Jetson Nano	Kinect	No Ejecuta	5.07
Ordenador portátil	Webcam fhd	10.08	30.20
Ordenador portátil	Kinect	10.37	30.02

Tabla 4: Resultado de rendimiento de los diferentes ordenadores utilizando YOLO y TINY-YOLO y las diferentes cámaras.

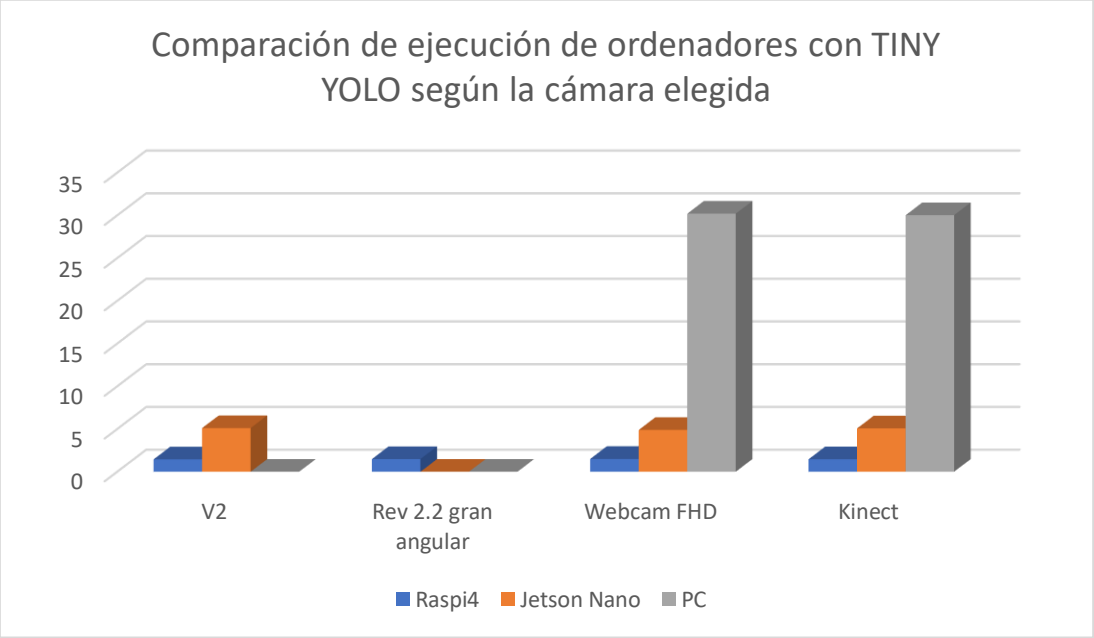

Gráfico 1: Comparación de ejecución de ordenadores con TINY-YOLO según la cámara elegida.

Hardware	Pros	Contras	Portable
Raspberry Pi 3 b+	- Compacta.	- Poca capacidad de procesamiento.	Si
Raspberry Pi 4	- Compacta.		Si
NVIDIA Jetson Nano	- Compacta. - Buena Gpu.	- No es suficientemente potente como para ejecutar la versión completa de YOLO.	Si
Ordenador portátil	- Buena capacidad de procesamiento.	- Por su peso y tamaño no es muy portable.	No

Tabla 5: Pros y contras de los ordenadores según los resultados obtenidos en el experimento anterior.

2.2.2. Hardware para la medición de distancias a los objetos

Otro factor para tener en cuenta como explicamos anteriormente es la distancia que hay a cada uno de los objetos de la imagen. Necesitamos saber aproximadamente que distancia existe a cada uno de los objetos de interés que explicamos en la imagen anterior.

Es preciso conocer la posición de los objetos si queremos indicar la posición mediante sonido espacial del objeto ya que aprovecharemos las ventajas del sonido 3d para indicar proximidad y localización.

Input	Pros	Contras
Cámara Depth	- Otorga distancia con los objetos con una precisión muy alta. - Es el hardware de mayor alcance	- No suelen ser muy pequeñas. - Si está basada en infrarrojos, puede tener problemas con las luces.
Sensores infrarrojos	- Preciso al medir objetos cercanos.	- El alcance es como mucho medio metro. - Necesitamos muchos de ellos para poder obtener los datos que necesitamos.
Sensores ultrasonido	- Funcionan bien al medir objetos cercanos.	- El alcance es aproximadamente 300 centímetros. - Necesitamos muchos de ellos para poder obtener los datos que necesitamos.
Cámara estéreo	- Permite calcular distancias en base a las imágenes de las dos cámaras. - Nos evitaríamos agregar otro tipo de hardware al proyecto.	- Alto coste computacional. - No son tan precisos como las cámaras Depth.
Sensores LIDAR de movimiento	- Nos otorga una imagen en tres dimensiones con las distancias que hay a todos los objetos. - Existen en formato pequeño.	- Pueden ser un poco molestas ya que llevan un pequeño motor que las hace girar.
Array de sensores LIDAR	- Nos otorga una imagen en tres dimensiones con las distancias que hay a todos los objetos visibles. - Existen en formato pequeño.	- Algunos son muy grandes y no son portables.

Tabla 6: Posibles inputs con sus pros y contras.

Hardware	Pros	Contras	Posibles inputs
Gorra	- Pueden adaptarse componentes grandes a ella.	- No todo el mundo lleva gorra, sería algo agregado para la persona.	Cámara Depth, Sensores infrarrojos, Sensores ultrasonido, Cámara estéreo, Sensores LIDAR de movimiento, Array de sensores LIDAR
Gafas	- Casi todas las personas invidentes las llevan. - Es una altura correcta para simular los ojos.	- Sólo pueden ir inputs pequeños en ellas.	Cámara Depth, Cámara estéreo, Sensores LIDAR de movimiento, Array de sensores LIDAR
Cinturón	- Permite una buena medición cuando se trata de medir objetos cercanos. - Puede ser bueno para colocar los sensores LIDAR.	- No todo el mundo lleva cinturón.	Cámara Depth, Sensores infrarrojos, Sensores ultrasonido, Cámara estéreo, Sensores LIDAR de movimiento, Array de sensores LIDAR
Guantes	- Permite palpar proximidades. - Es muy intuitivo a la hora de caminar a ciegas.	- Es un agregado que no necesariamente lleva una persona. - No sirve para obtener una medición general de distancia.	Sensores infrarrojos, Sensores ultrasonido.

Tabla 7: Posible hardware donde colocar los inputs.

2.2.3. Software para la obtención de datos

Para el reconocimiento de imágenes vamos a utilizar un framework llamado Darkflow, el cual nos permite aplicar las distintas versiones de YOLO de manera eficiente con Tensorflow.

Además, utilizaremos la cámara Kinect ya que incorpora ambos sensores en una misma y nos otorga un nivel de precisión bastante bueno.

Cabe aclarar que Intel Realsense para un modelo final es la que mejor se adapta a lo que estamos buscando, pero Kinect para nuestro modelo experimental funciona correctamente.

Por otra parte, para controlar la Kinect utilizaremos la librería Libfreenect [6] que nos permite controlar la obtención de los siguientes frames de manera más precisa que la librería oficial, ya que podemos controlar de manera exacta cuándo queremos obtener el siguiente frame y también nos lo devuelve en un formato agradable para tratar dicha imagen.

2.3. Fase de procesamiento

En esta fase, nos vamos a centrar en la forma en que vamos a guardar los datos de cada objeto detectado.

Primero vamos a detectar los objetos en la imagen, de ellos sacaremos su bounding box con el que obtendremos el punto central de cada uno (centerX,centerY).

A continuación cada cierto tiempo, vamos a ejecutar en otro hilo una función que se encargará de obtener la distancia mediante la cámara Depth del punto central (centerX,centerY) para así guardarla como la distancia hacia el objeto. El valor que obtenemos es en milímetros, por ende, lo dividiremos por 10 para que esté en centímetros.

La información de cada objeto la guardaremos de la siguiente manera:

infoObjeto = [angulo,distancia,nombre]

Este array representa la información de un solo objeto, de manera que necesitaremos otro array para guardar la información de todos los objetos de la siguiente manera:

arrayObjetos = [infoObjeto1, infoObjeto2, infoObjeto3, …, infoObjetoN]

En cuanto a la información del objeto ya tenemos su distancia y su nombre, ya que YOLO nos entrega un array con la siguiente información:

objetoYolo=[bounding_box,probabilidad,nombre]

Ahora necesitamos obtener el ángulo que tiene el objeto en la imagen y lo calcularemos de la siguiente manera:

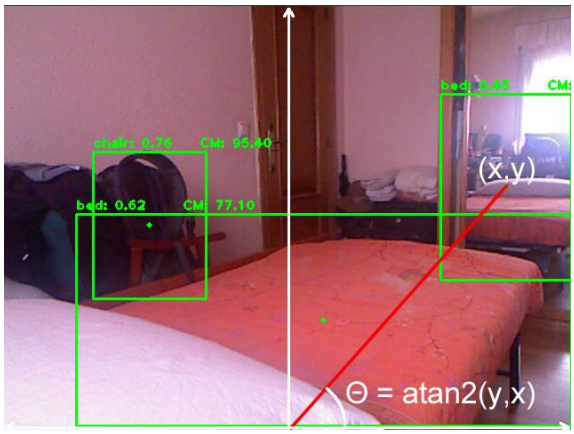

Figura 18: Explicación de cómo obtener el ángulo del centro del objeto.

Como puede verse utilizamos la función del Arcotangente de dos parámetros para calcular dicho ángulo.

Podemos observar que los valores de x e y que nos devuelve el punto central del objeto no tiene en cuenta que el origen esté en el centro de la pantalla, para ello tenemos que ajustarlo al verdadero centro de la pantalla.

Utilizamos una imagen de 640 píxeles por 480 píxeles lo que quiere decir que el centro de la pantalla está en 320 pixeles por 240 píxeles.

Para obtener los verdaderos puntos de nuestro objeto en base a dicho centro, vamos a tener en cuenta también que la posición (0,0) de la pantalla está en la parte superior izquierda de la pantalla y el eje x crece hacia la derecha, pero el eje y crece hacia abajo.

Entonces el nuevo punto será:

$$nuevoX = centroPantallaX – antiguoX$$

$$nuevoY = máximoY - antiguoY$$

De esta manera ya tenemos el punto ajustado para poder aplicar la función Arcotangente para dos parámetros.

Una vez tengamos todos los objetos en su array como explicamos anteriormente, vamos a poder ordenarlos de dos maneras.

Ordenarlos por su ángulo, de manera que el sistema los comunique según su ángulo de inclinación. U ordenarlos por su distancia, de manera que la comunicación sea primero los más cercanos y luego los más lejanos.

Por último y no menos importante es seleccionar el umbral por el que comenzaremos a nombrar los objetos por su nombre y no por el nombre objeto (ya que queremos decir lo que es con precisión y no decir que es algo que luego no lo sea).

2.3.1. Cálculo del umbral de nombramiento de objetos

Al obtener los objetos con YOLO existe un parámetro llamado threshold, el cuál filtra los objetos detectados por debajo de ese valor. Por ejemplo, si detecta que hay una persona con una probabilidad del 30%, entonces esta persona no aparecerá en nuestra imagen. Normalmente, este tipo de casos se da con personas que están muy lejos en la imagen, lo cual no nos importa mucho, pero ¿Qué pasa con esos objetos que hay delante de nosotros y que no sabe identificarlos? En este caso necesitamos que nos lo muestre, aunque su probabilidad sea baja, así que he decidido colocar dicho threshold en 0,1 para que muestre absolutamente todo.

Este threshold puede traernos problemas, ya que puede detectar dos objetos donde sólo hay uno, por cómo hemos explicado que funciona YOLO. Pero lo importante es que nos detecta todos y cada uno de los objetos en la imagen.

Este escenario nos plantea un nuevo problema ¿Qué objetos nombramos y cuáles no? Hemos dejado pasar todos los datos que vienen de la red neuronal sin filtrar nada, entonces el filtro lo tendremos que dar nosotros.

Para ello vamos a determinar un umbral el cual nos servirá para decir si probabilidad-objeto > umbral, entonces nombro el objeto, si probabilidad-objeto <= umbral, entonces simplemente digo objeto. Pero ¿Cuál es dicho umbral y cómo lo voy a calcular?

Para responder a esa pregunta voy a utilizar una base de datos de 10 imágenes de calles con personas y objetos. En total hay para clasificar unos 400 objetos que serán clasificados manualmente mediante un software que he diseñado en dos clases: Correcto e incorrecto.

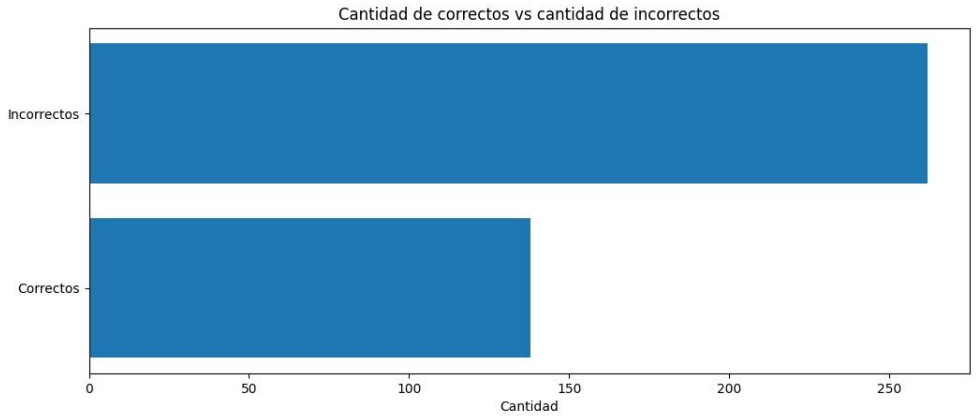

Gráfico 2: Gráfico de cantidades de casos clasificados correctamente versus incorrectamente

Los casos incorrectos son 262 objetos, mientras que los casos correctos son 138.

Ahora todo se reduce a un problema de aprendizaje supervisado. Tenemos las dos clases ya armadas con los ejemplos que el clasificador ha clasificado correcta e incorrectamente. Ahora llega el momento de nuestro umbral, el cual lo haremos variar entre 0,3 y 0,8 con un salto de 0,02 y para cada interación calcularemos su matriz de confusión para poder determinar el rendimiento de nuestra clasificación según dicho umbral.

Mostraremos las matrices de confusión que se van obteniendo de cada umbral seleccionado con los datos de accuracy, recall y precisión.

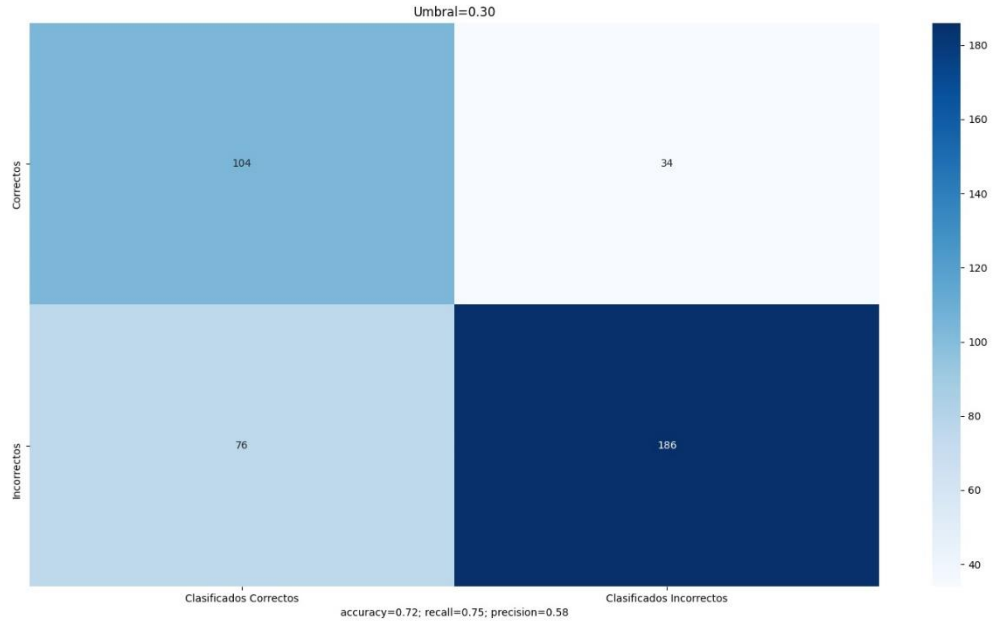

Figura 19: Matriz de confusión para el umbral 0.30

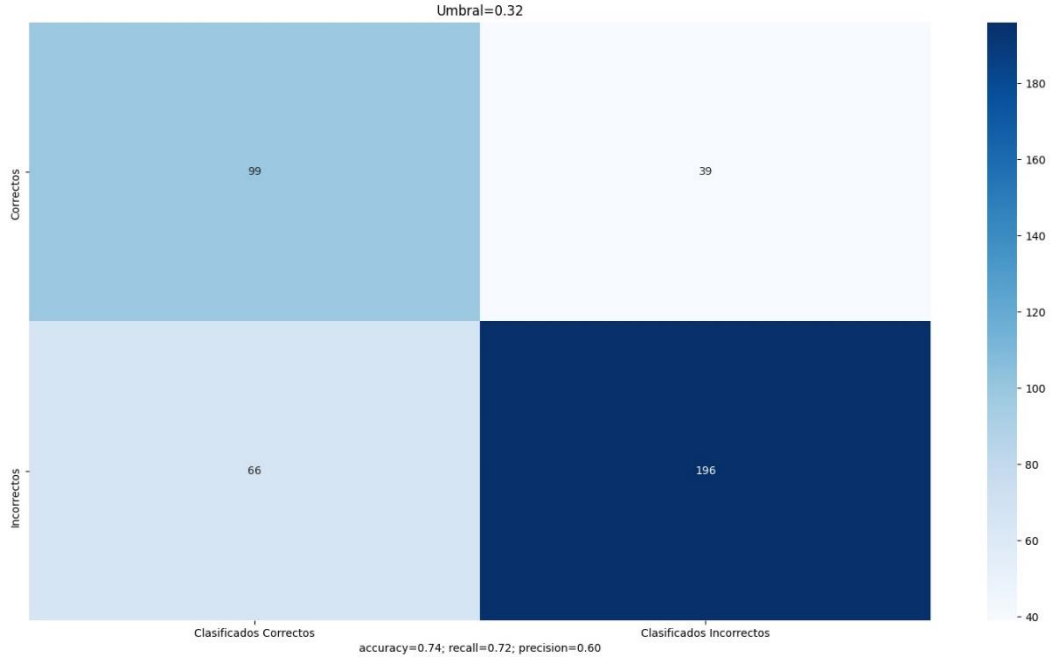

Figura 20: Matriz de confusión para el umbral 0.32

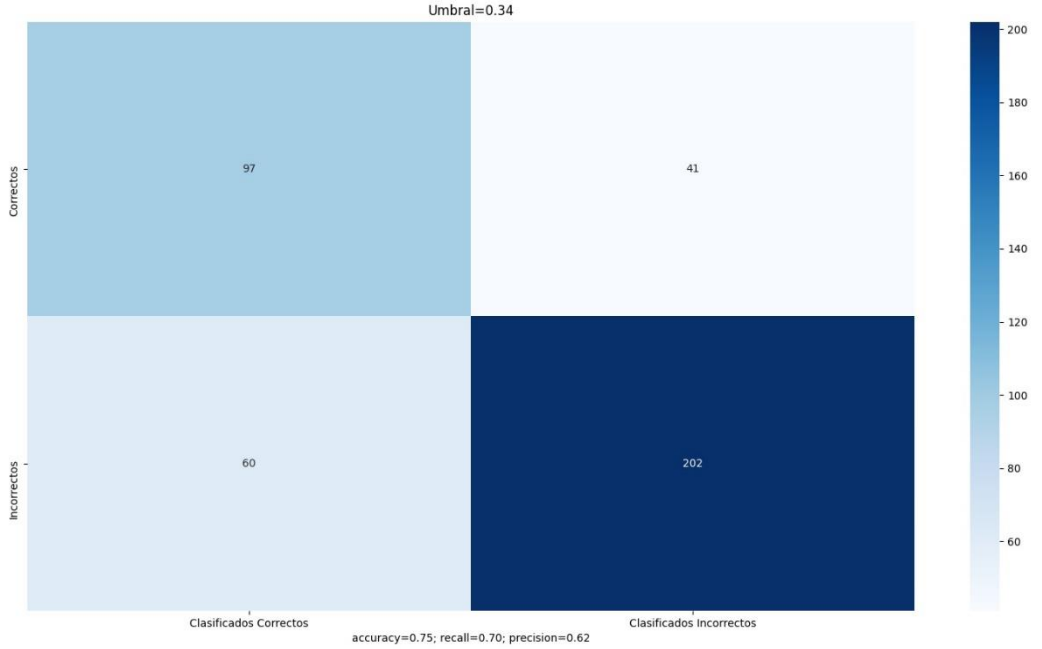

Figura 21: Matriz de confusión para el umbral 0.34

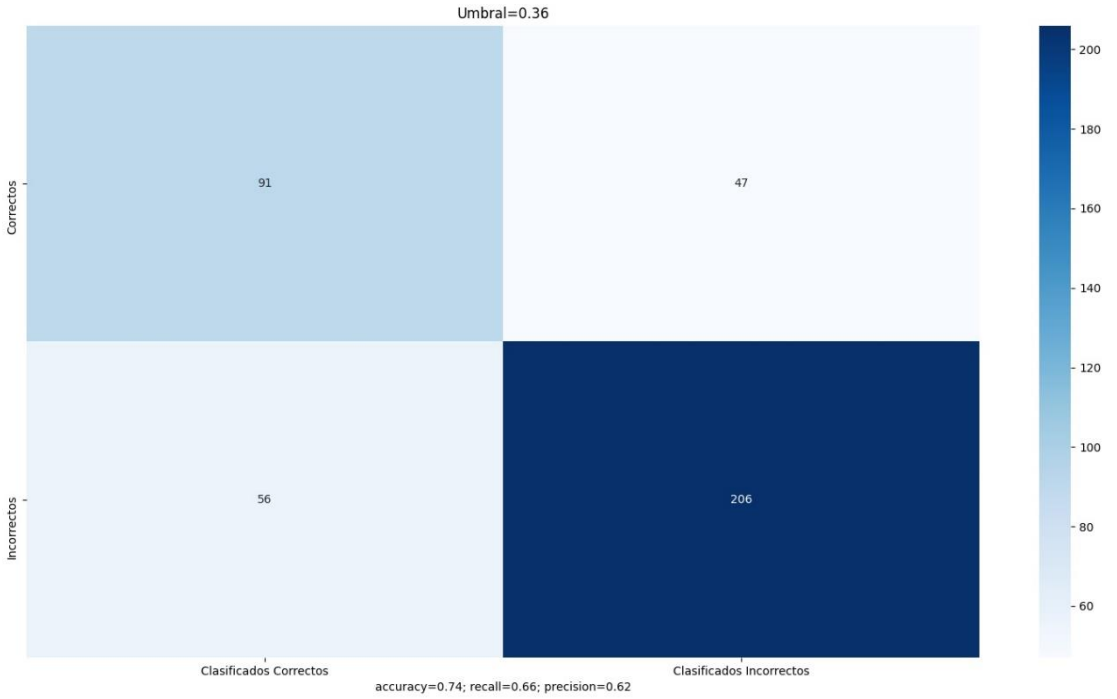

Figura 22: Matriz de confusión para el umbral 0.36

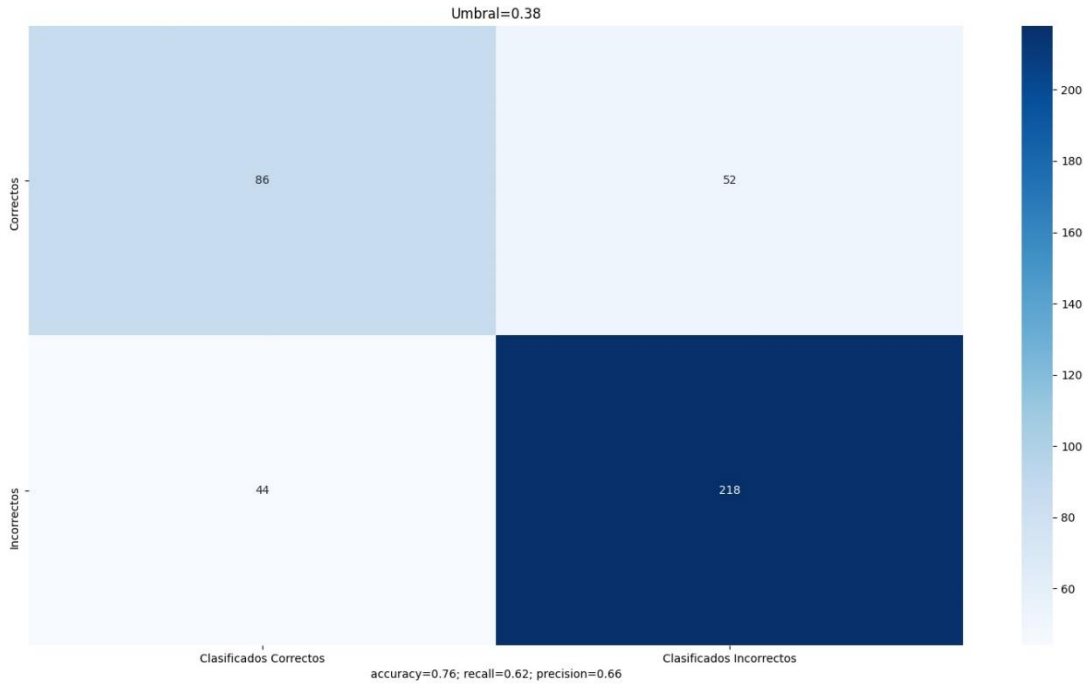

Figura 23: Matriz de confusión para el umbral 0.38

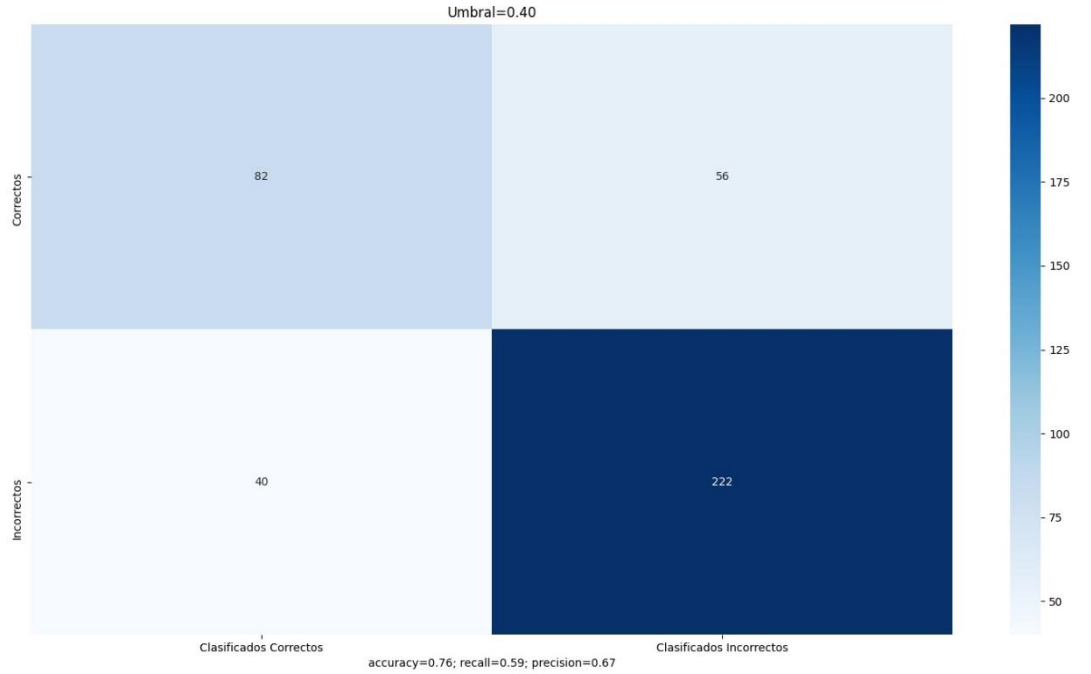

Figura 24: Matriz de confusión para el umbral 0.40

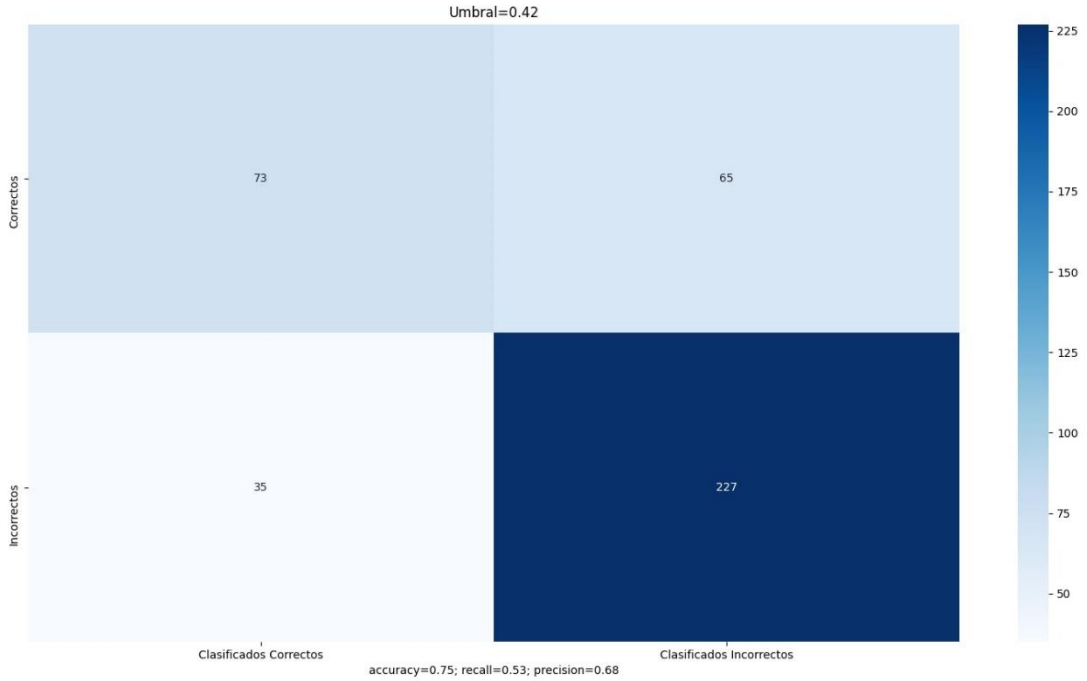

Figura 25: Matriz de confusión para el umbral 0.42

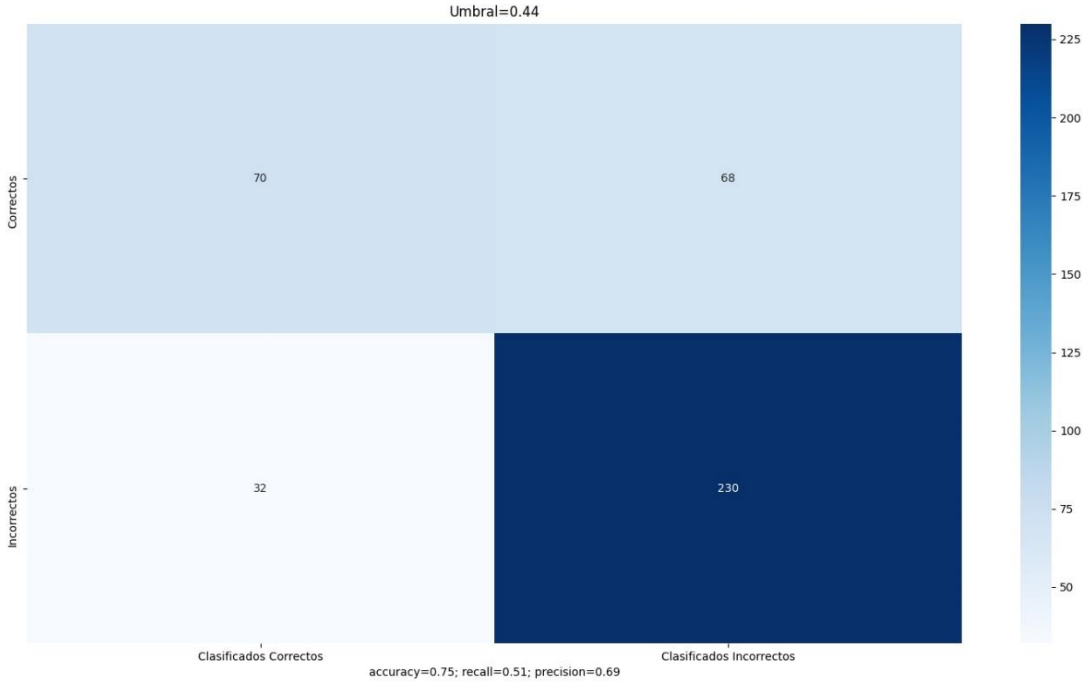

Figura 26: Matriz de confusión para el umbral 0.44

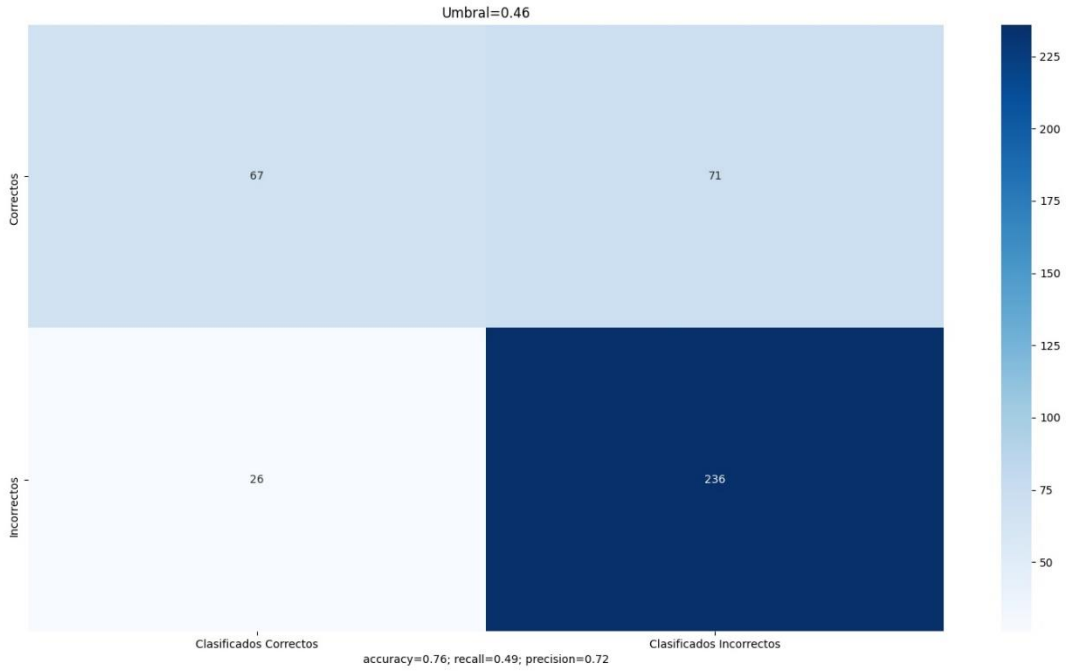

Figura 27: Matriz de confusión para el umbral 0.46

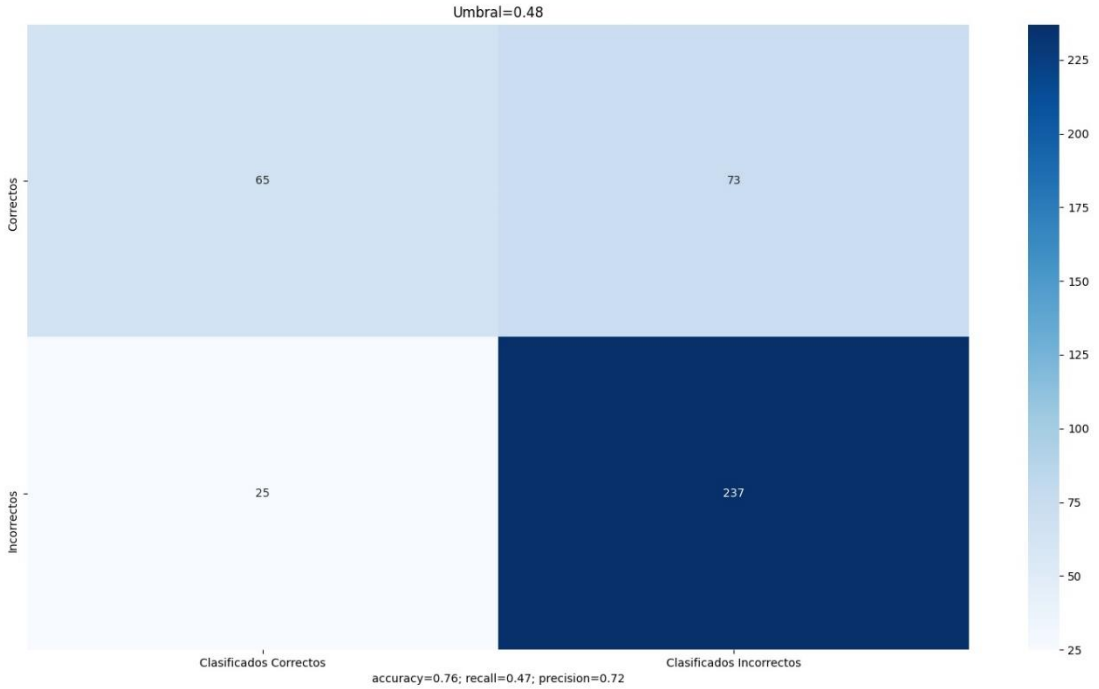

Figura 28: Matriz de confusión para el umbral 0.48

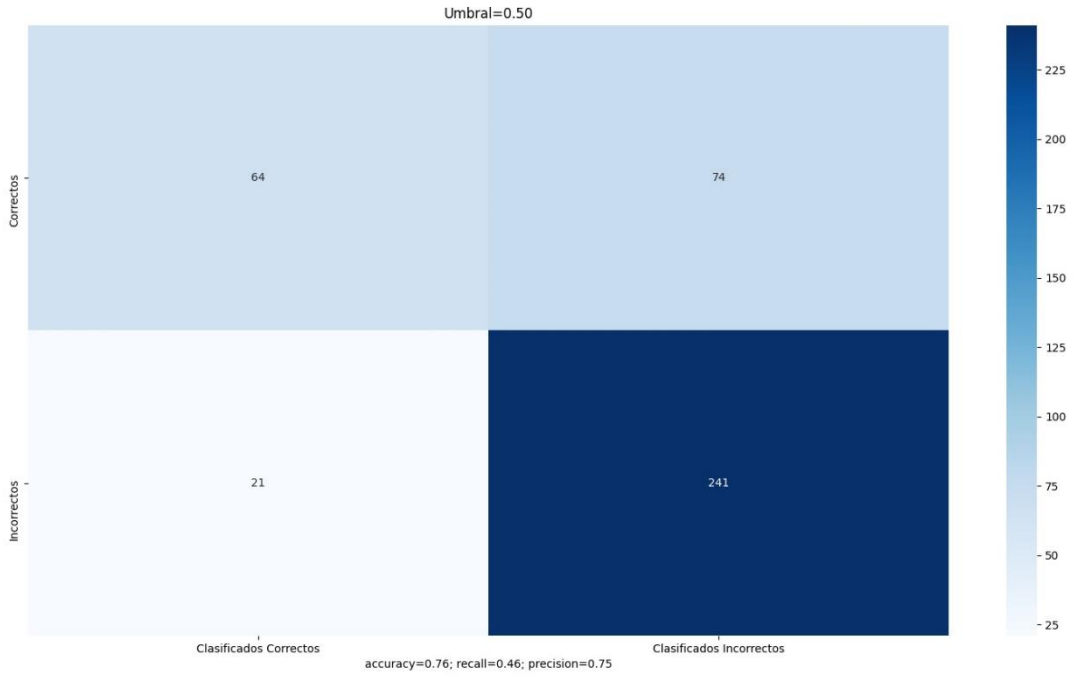

Figura 29: Matriz de confusión para el umbral 0.50

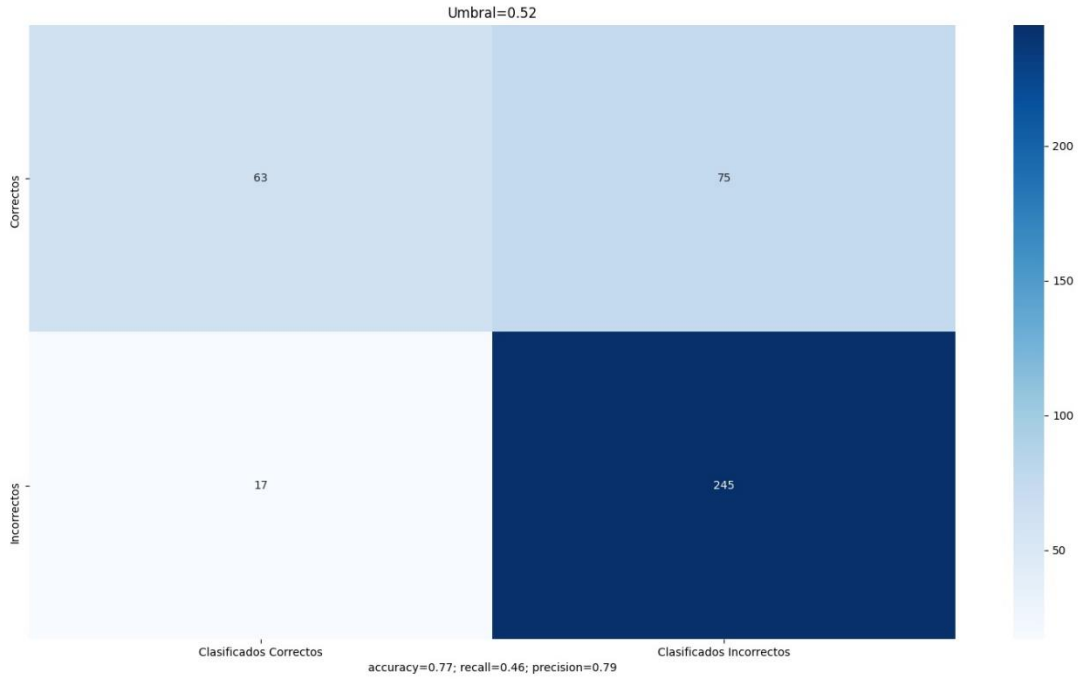

Figura 30: Matriz de confusión para el umbral 0.52

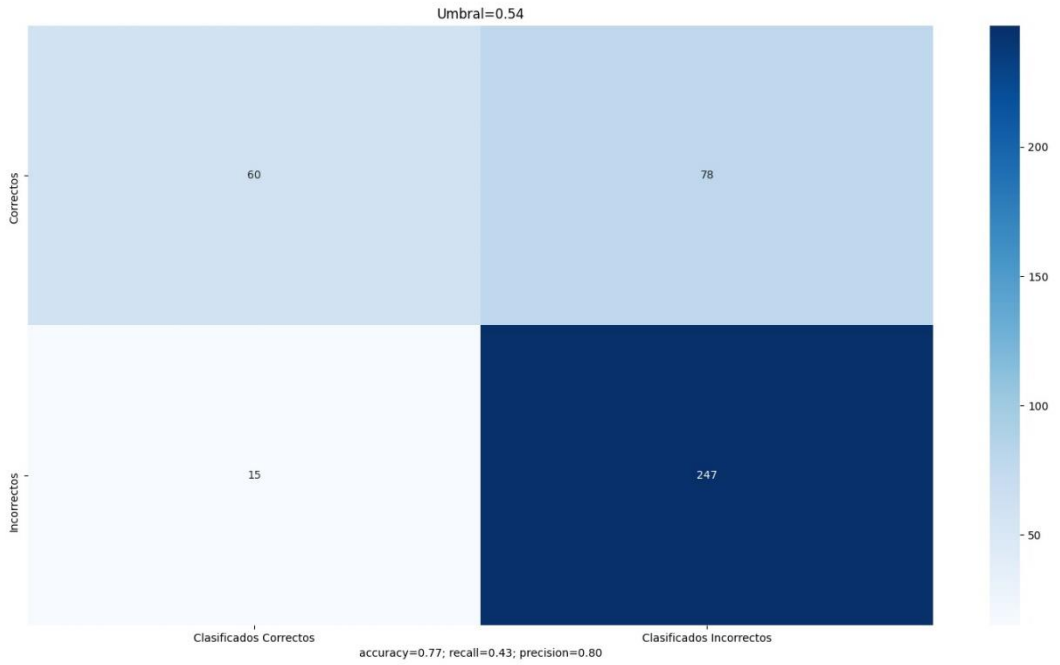

Figura 31: Matriz de confusión para el umbral 0.54

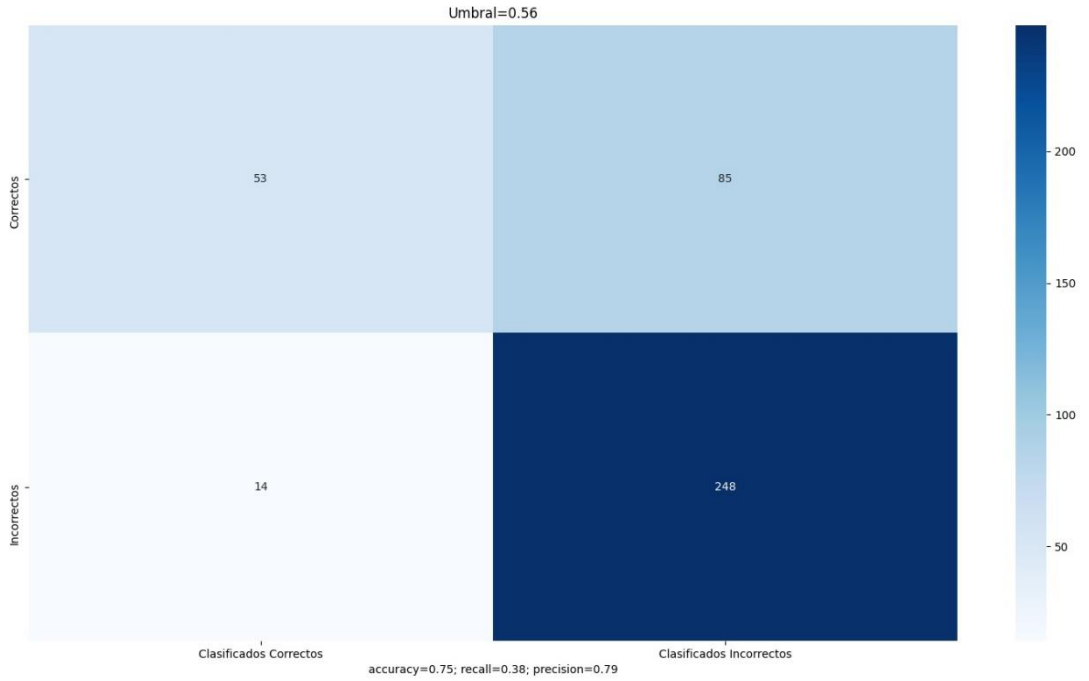

Figura 32: Matriz de confusión para el umbral 0.56

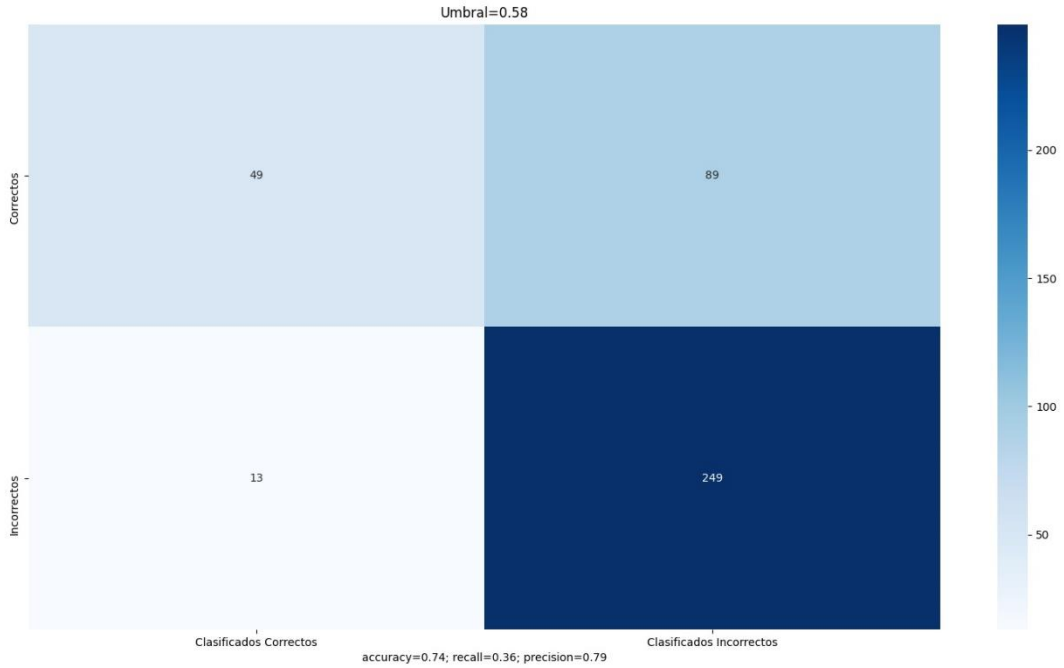

Figura 33: Matriz de confusión para el umbral 0.58

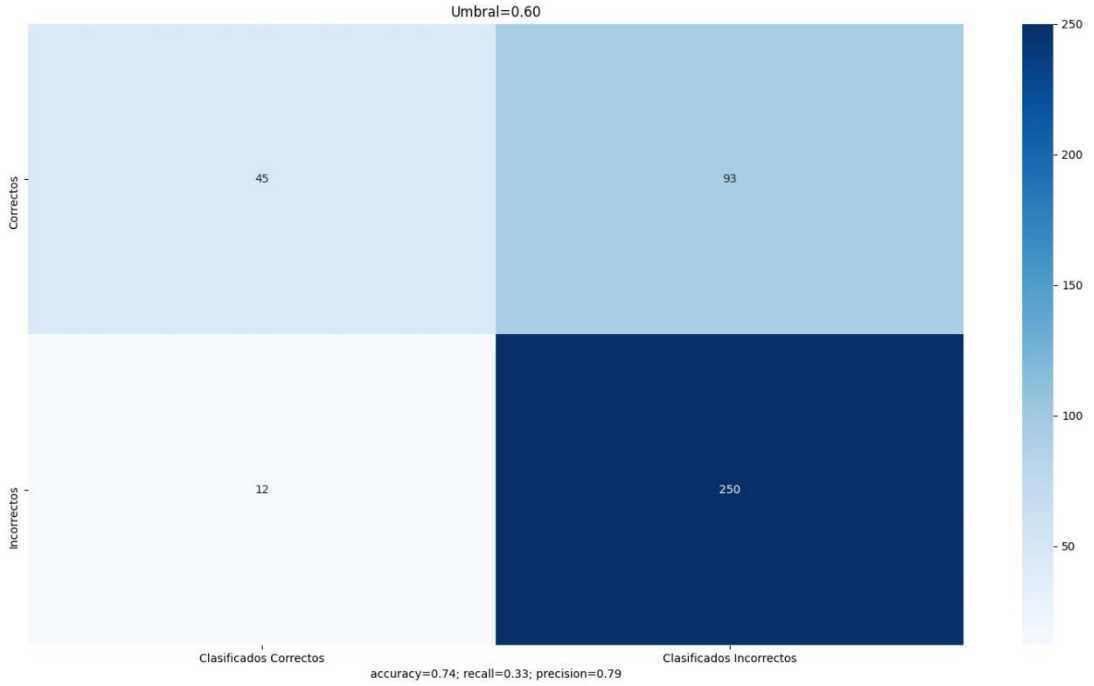

Figura 34: Matriz de confusión para el umbral 0.60

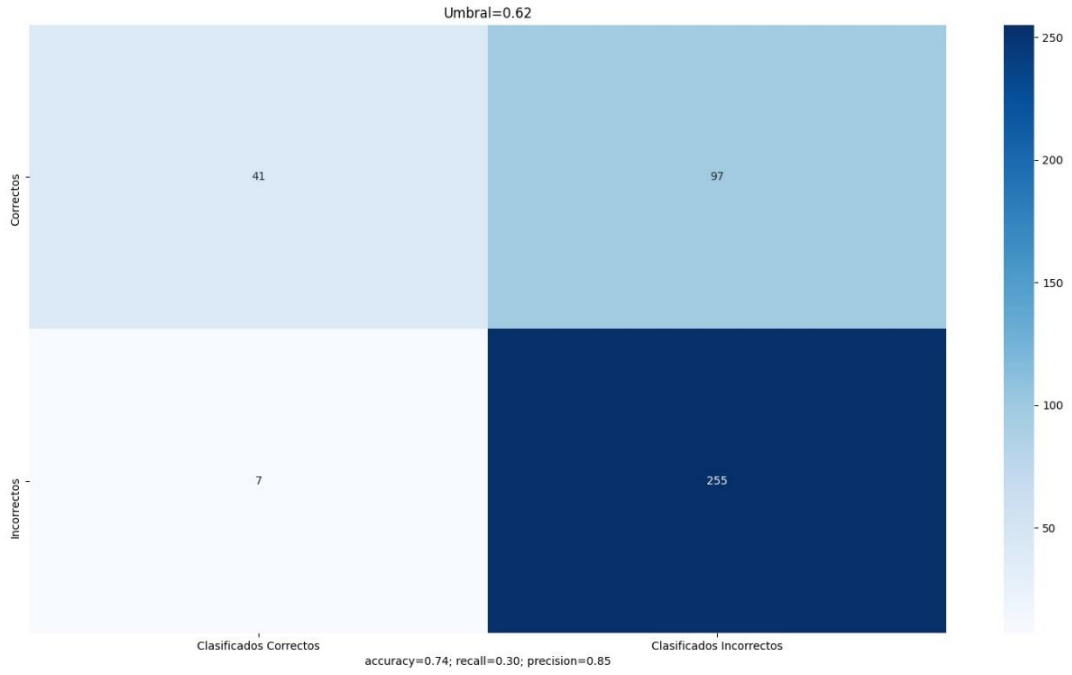

Figura 35: Matriz de confusión para el umbral 0.62

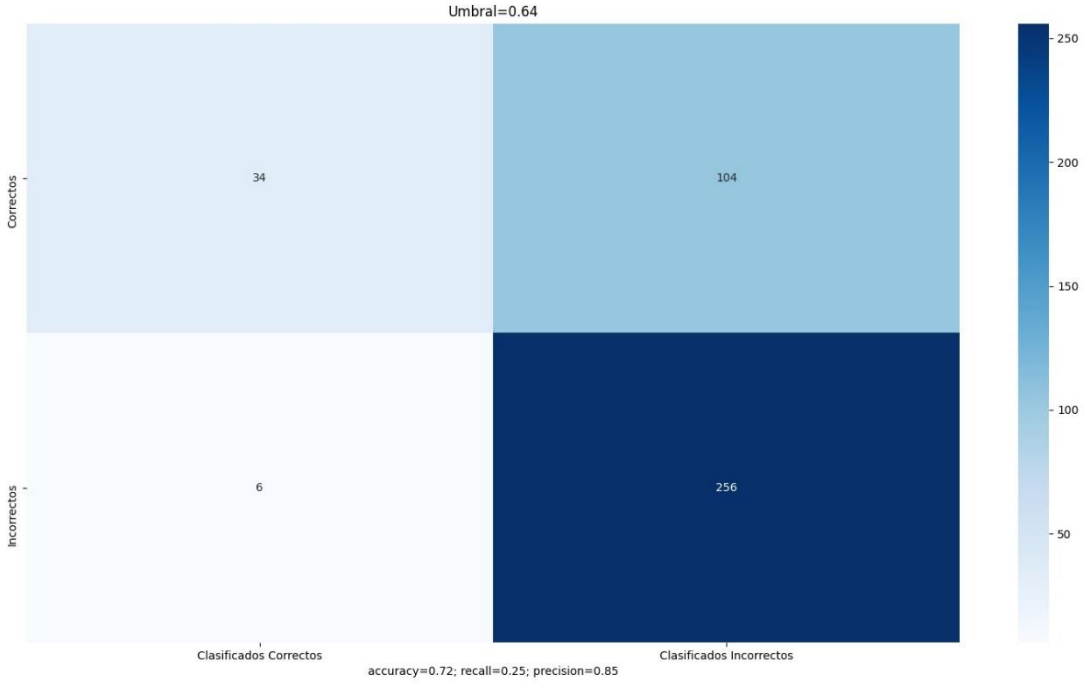

Figura 36: Matriz de confusión para el umbral 0.64

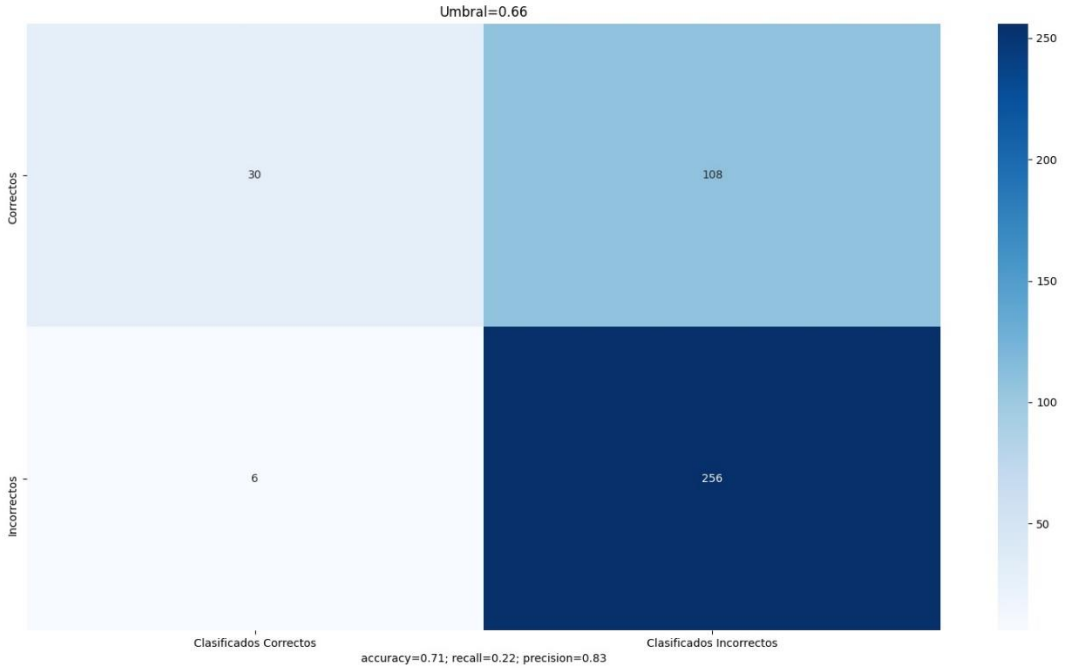

Figura37: Matriz de confusión para el umbral 0.66

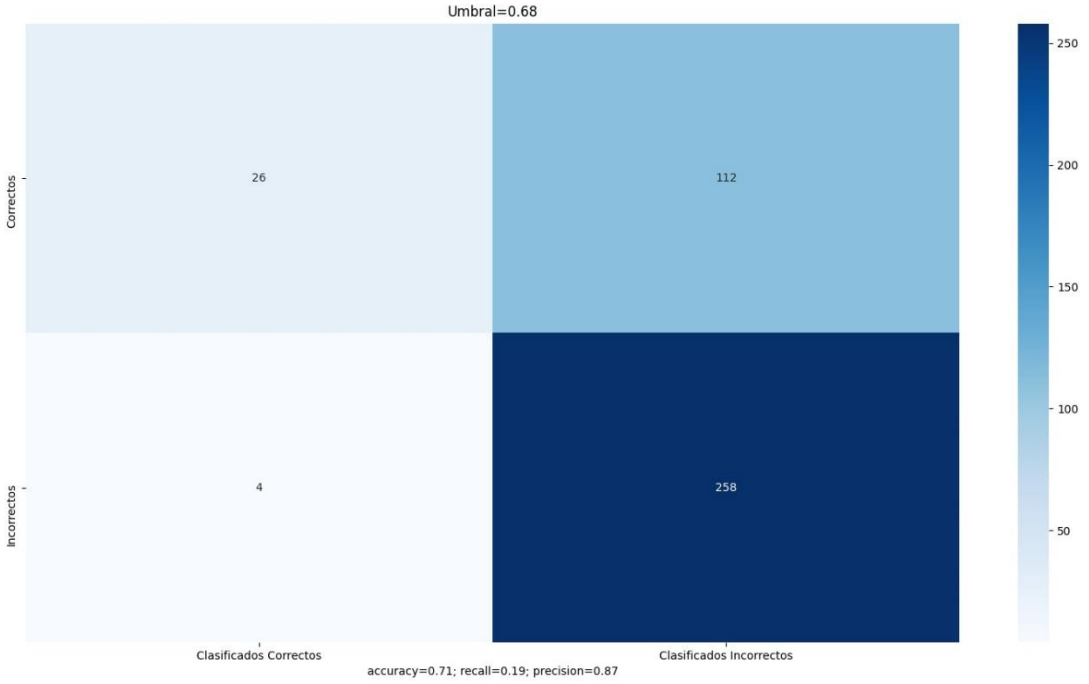

Figura38: Matriz de confusión para el umbral 0.68

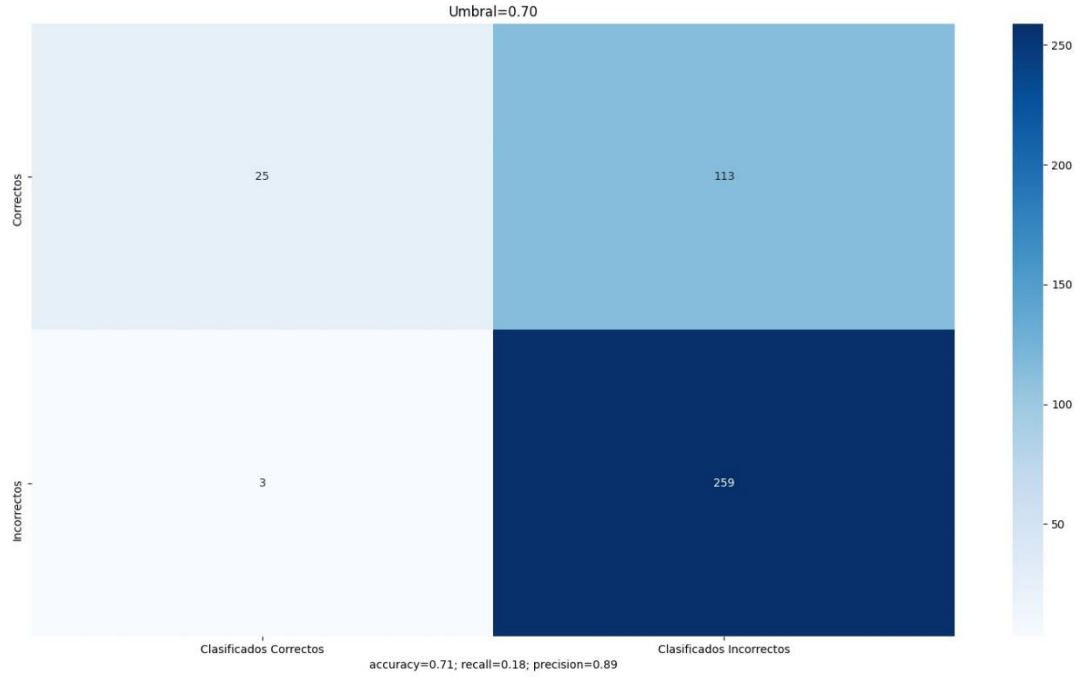

Figura 39: Matriz de confusión para el umbral 0.70

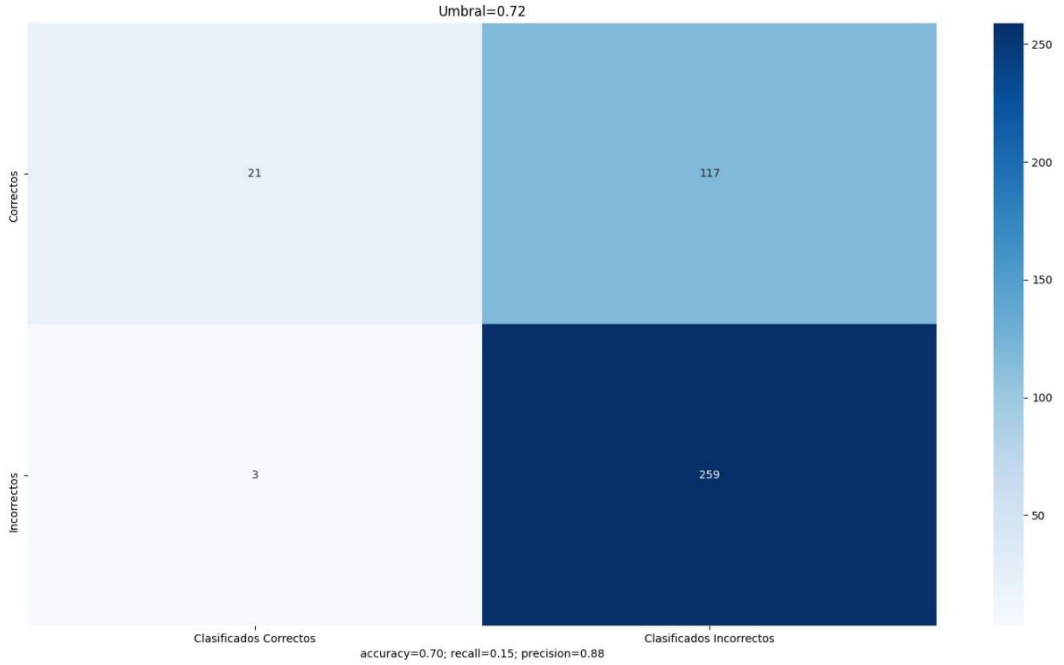

Figura 40: Matriz de confusión para el umbral 0.72

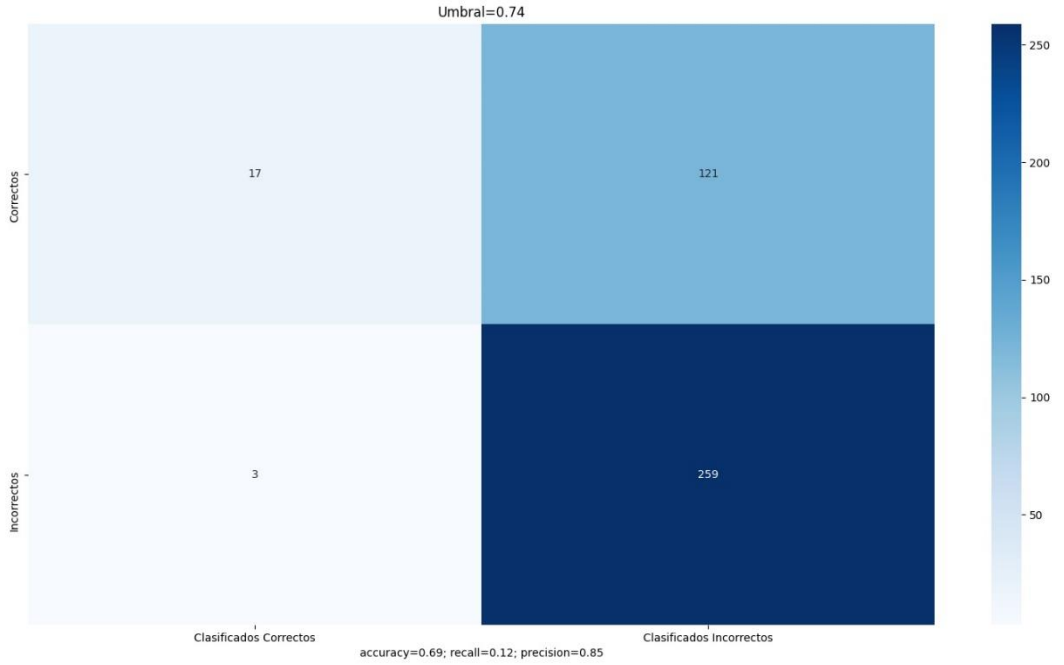

Figura 41: Matriz de confusión para el umbral 0.74

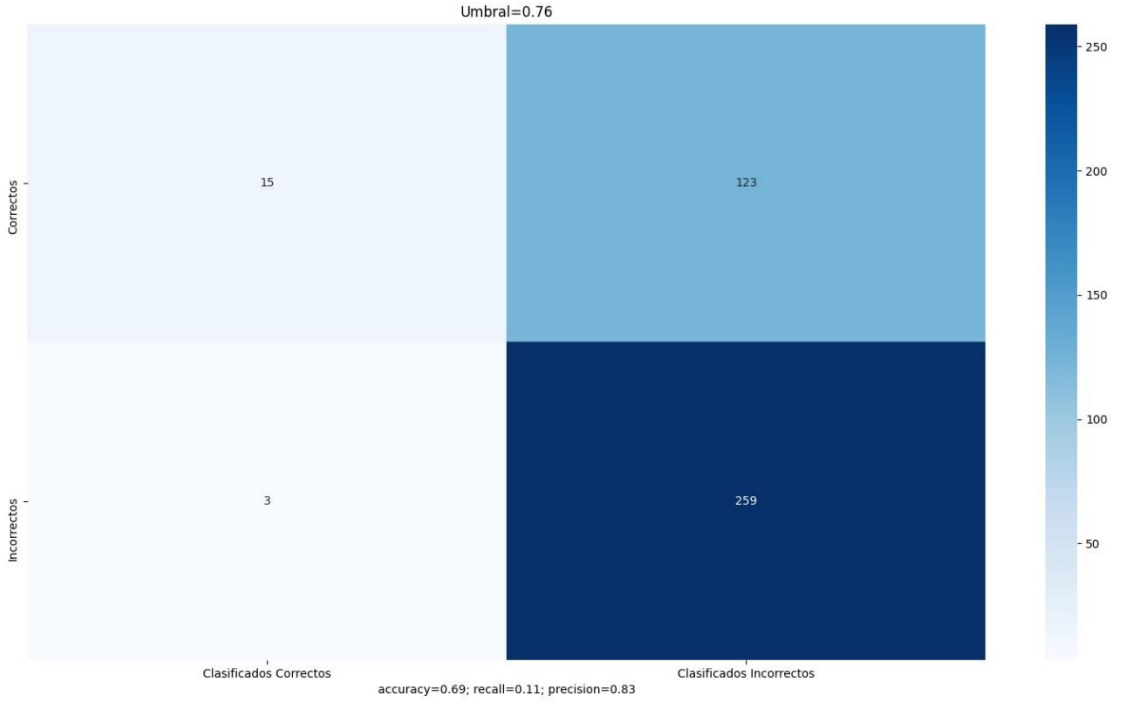

Figura42: Matriz de confusión para el umbral 0.76

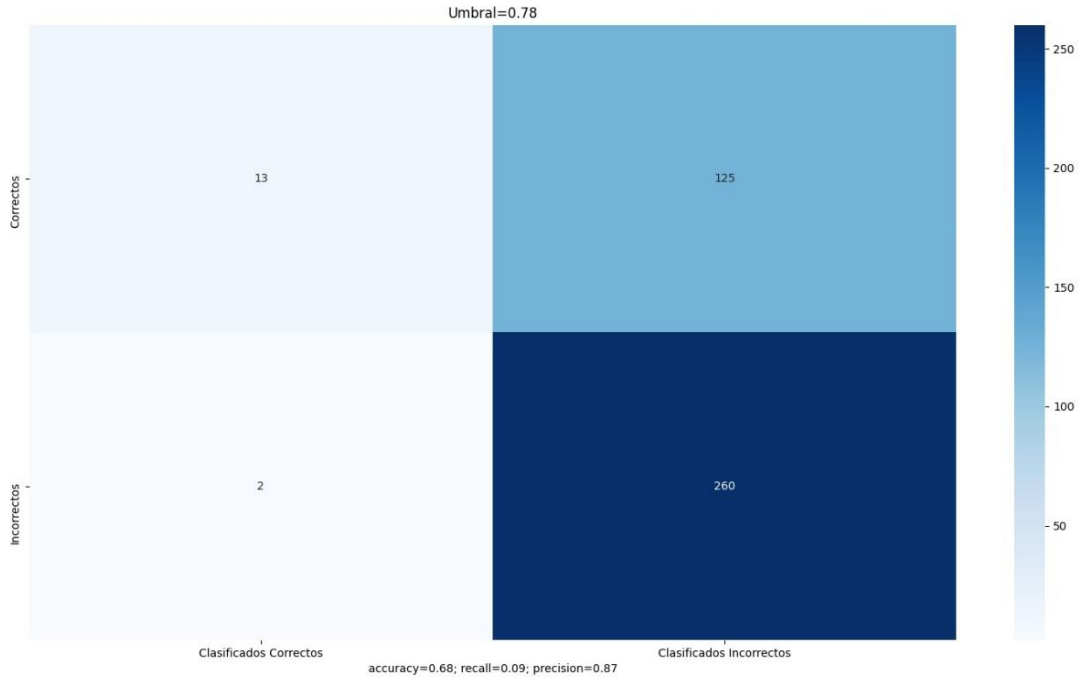

Figura 43: Matriz de confusión para el umbral 0.78

He decidido armar un histograma para poder ver con más claridad por cada umbral, el accuracy, recall y precisión del modelo.

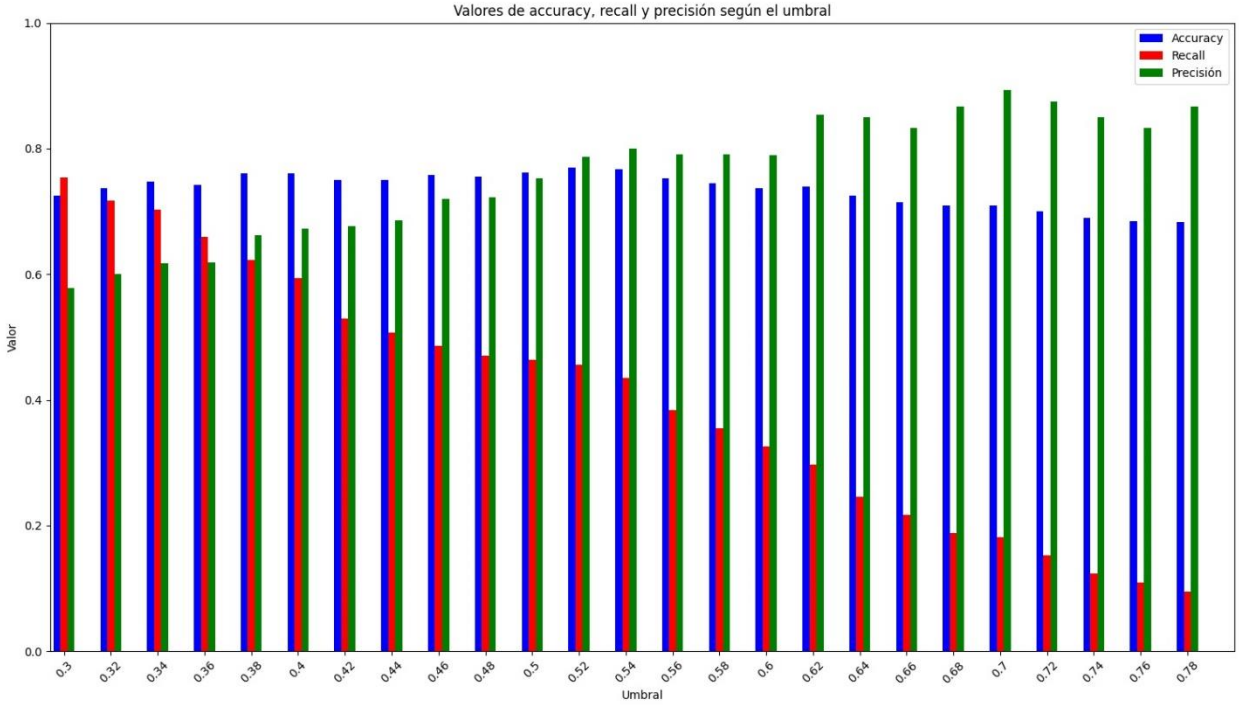

Gráfico 3: Histograma que compara cada umbral con sus valores de accuracy, recall y precisión.

Para aclarar las siguientes decisiones, vamos a repasar un poco los conceptos de accuracy, recall y precisión.

Accuracy es el porcentaje de casos del modelo que he acertado en la selección, determina que tan correcto es el modelo. Por ende, buscamos un accuracy alto en nuestro caso.

Recall es la cantidad de objetos que son correctos, que según nuestro umbral hemos clasificados correctamente. Hay que tener en cuenta que, si nuestro modelo clasifica un solo objeto correcto como correcto y los demás como incorrectos, el recall va a ser muy alto.

Precisión es la cantidad de predicciones correctas, ya sea con objetos inicialmente clasificados correctamente como incorrectamente.

Lo que vamos a intentar conseguir, es el modelo con mayor accuracy y precisión, pero cuidando de que el recall no sea muy bajo, ya que queremos minimizar la cantidad de falsos positivos.

Según los datos obtenidos, podemos seleccionar el 0,4 como un umbral correcto.

He decidido obtener también un boxplot de los casos correctamente clasificados y los que son incorrectamente clasificados por TINY-YOLO.

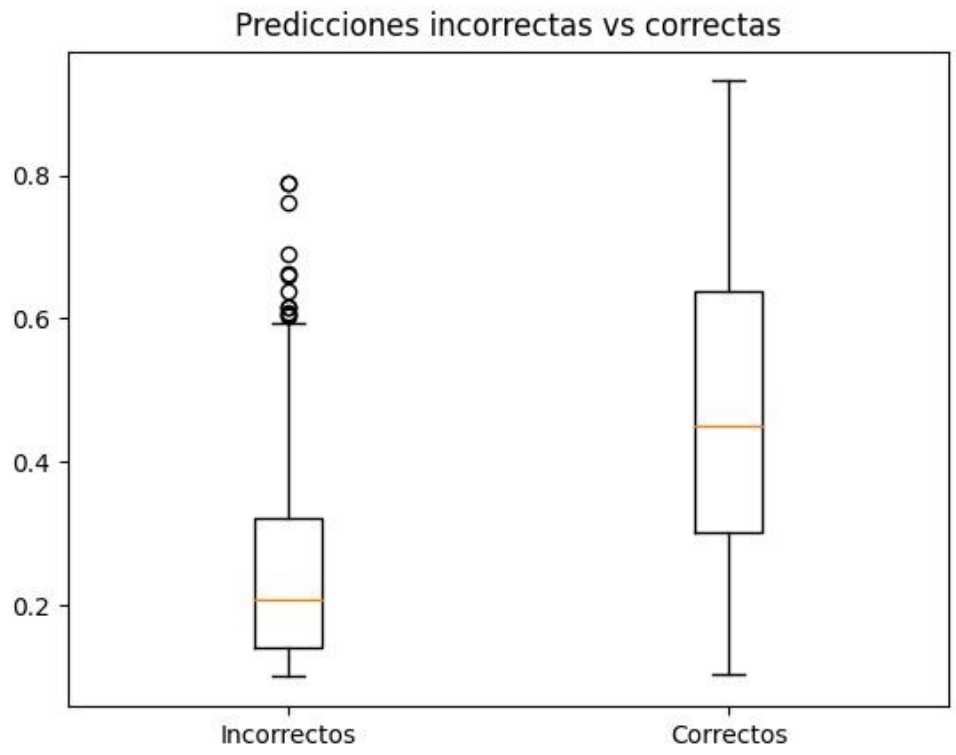

Gráfico 4: Boxplot que muestra los valores de los objetos clasificados correcta e incorrectamente.

Podemos ver que los casos incorrectos de media están cerca del 0,2 de threshold, mientras que los correctos están cerca del 0,45. También se puede ver que los casos correctos ocupan todo el espectro de valores, ya que tiene mínimos en el 0,1 (serán objetos lejanos que ha detectado correctamente, pero con poca probabilidad) y el 0,9 como máximo. También podemos obtener un resultado valioso en esta gráfica, que son los outliers de los casos incorrectos, ya que ha detectado con mucha precisión objetos incorrectamente, pero como la definición de outlier lo dice, son casos aislados. Podemos afrontar estos pequeños errores, ya que, si queremos hacer que el programa sólo nos nombre objetos que detecte al 100% correctamente, tendríamos que poner el umbral arriba del 0,8 y cometeríamos muchos falsos negativos al clasificar como incorrectos objetos que el programa los ha detectado correctamente. Podemos observar que, dentro de los valores normales de los incorrectos, el máximo está cerca del 0,6 (quizás un pelín más abajo), lo que lleva a pensar que si queremos minimizar al máximo los falsos positivos como hemos charlado con nuestro entrevistado, tendríamos que seleccionar el umbral 0,62.

Este umbral tiene un recall muy bajo, pero tiene una precisión muy alta y también sucede esto con el accuracy. El recall es muy bajo ya que al elevar tanto el umbral, estamos clasificando muchos objetos correctos como incorrectos. Pero para nosotros y el usuario eso no es ningún problema, ya que el objeto se va a comunicar igualmente sólo que no se lo nombrará con su clase predicha. De esta manera reducimos drásticamente la posibilidad de decirle al usuario que está viendo una persona cuando realmente ve una farola.

2.4. Fase de comunicación con el usuario

Nuevamente hay muchas posibilidades a analizar para poder comunicar al usuario las decisiones tomadas en la etapa anterior. Analizaremos cada una de ellas, sus pros y sus contras y trataremos de elegir la que mejor se adapte a las necesidades y la más adaptable también a nuestro proyecto.

2.4.1. Hardware para la comunicación con el usuario

Medio	Pros	Contras
Voz	- Si se procesa bien la información y se logra enviar la información justa es la menos molesta para la persona.	- Muy molesto si se vuelve repetitivo o da indicaciones cada pocos segundos.
Pitido	- Sirve para indicar proximidad hacia los distintos objetos. Como funciona el sensor de aparcamiento de los coches pita poco al principio y a medida que nos acercamos cada vez mas	- La información que se le da al usuario no es muy precisa. - Puede ser muy molesto y no permitir escuchar el entorno a la persona.
Vibraciones	- También indica proximidad a los objetos mediante vibraciones.	- La información que se le da al usuario no es muy precisa. - Puede generar una sensación molesta cuando se deja de utilizar.

Tabla 8: Posibles outputs de sonido.

Hardware	Pros	Contras
Auriculares conducción ósea	- Se puede enviar sonidos al usuario sin que éste pierda audición del entorno.	- La calidad del sonido no es la misma que la de un auricular normal.
Auriculares normales	- Se pueden emitir sonidos con una buena calidad de audio.	- El usuario ve su audición reducida.

Tabla 9: Posibles hardwares para reproducir los outputs.

2.4.2. Software para la comunicación con el usuario

Para probar los distintos hardwares, vamos a utilizar un software que he realizado especialmente para ello utilizando las librerías opencv y Libaudioverse [7].

En dicho software se pretende que el usuario agudice primero su oído repitiendo la evaluación unas 10 veces, para luego comenzar con la evaluación definitiva que consta de escuchar un sonido proveniente desde una distancia y ángulo aleatorio y seleccionar en un plano 2d visual representativo de donde ha escuchado venir el sonido y a que distancia.

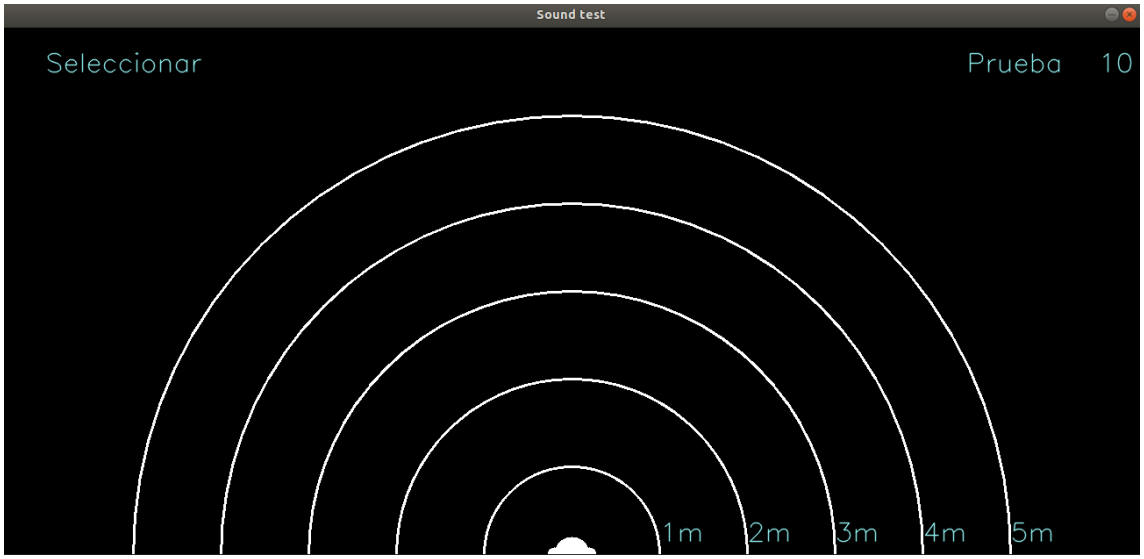

Figura 44: Software para la prueba de hardware sin empezar.

La dirección es fácil de percibir con la función hrtf pero la distancia es algo mucho más complicado, ya que la distancia se nota según la atenuación del sonido.

O sea, si se escuchan varios sonidos sucesivos alejándose nos daremos cuenta de que se está alejando (o acercando), pero si los escuchamos cada cierto tiempo nuestro cerebro debe generar un mapa de distancias en su cabeza basados en los sonidos que va escuchando. Por esto mismo es que he decidido realizar una fase de entrenamiento con diez intentos antes de comenzar con el ejercicio final.

En la interfaz visual puede verse en la parte inferior central nuestra cabeza, luego cada 100 pixeles hay círculos concéntricos simulando ser 1 metro.

Tiene dos fases, una de escuchar y otra de seleccionar. Primero escuchamos el sonido, inmediatamente que ha terminado de oírse, empieza la fase de selección, en la que vamos a hacer click en alguna parte del mapa según lo que hayamos oído.

Para medir el rendimiento de los tipos de hardware vamos a utilizar el tiempo que ha tardado el sujeto en contestar, el error angular y el error de distancia con respecto al punto original.

En la interfaz una vez que seleccionemos un punto, aparecerá un punto verde que indica el punto real de emisión del sonido y en rojo el punto que hemos seleccionado.

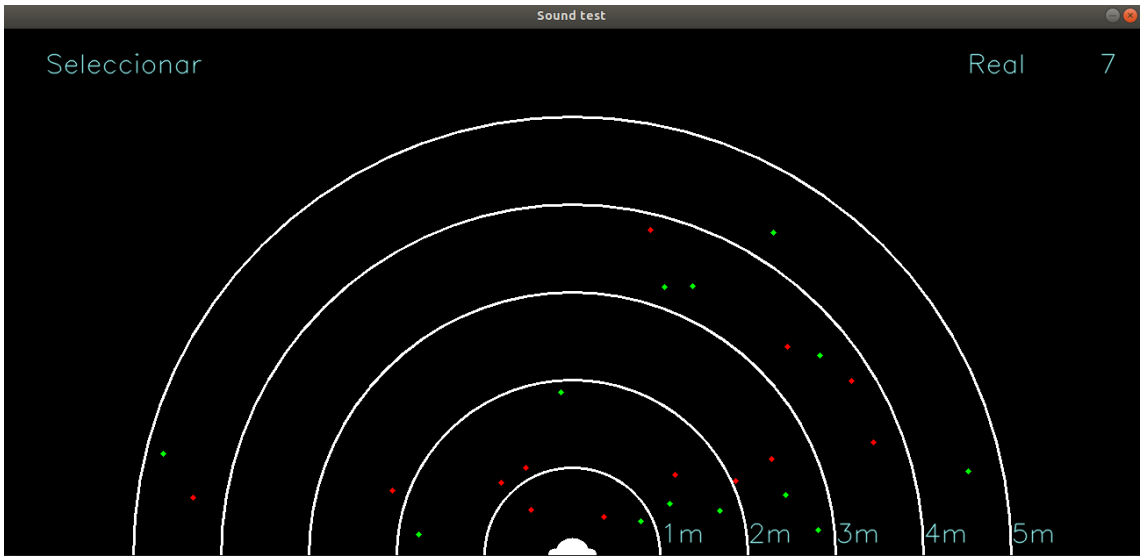

Figura 45: Software para la prueba del hardware en funcionamiento.

Por cuestiones públicamente conocidas, no he podido realizar esta prueba, ya que pretendía medir cuán distinto se percibe el sonido con los auriculares de conducción ósea con respecto a los normales, ya que los auriculares normales no son una opción válida, ya que le quita audición e información sobre el entorno al usuario.

Un ejemplo de ejecución de dicho software nos muestra los errores que se calculan por cada usuario que testea los cascos.

```
Final Data: [((733, 522), (697, 556), 1.0688722133636475), ((450, 457), (431, 38
1), 1.7110788822174072), ((637, 292), (638, 459), 1.064035177230835), ((646, 521
), (665, 522), 1.2817885875701904), ((721, 539), (721, 560), 1.0589604377746582)
, ((253, 588), (223, 436), 1.4968748092651367), ((290, 383), (193, 520), 2.79216
74251556396), ((351, 534), (451, 335), 1.067476749420166), ((842, 341), (706, 37
2), 1.2790184020996094), ((650, 410), (678, 477), 1.2831480503082275)]
Error de distancia medio: 0.27 metros
Error angular medio: 57.31º
Tiempo de resolución medio: 1.41 segundos
```

Figura 46: Ejemplo de resultados obtenidos en una prueba del software.

Podemos observar que obtenemos por cada prueba, el error en distancia, el error angular y el tiempo medio en que tarda el usuario en tomar una decisión en base al sonido que ha escuchado.

Resume del proyecto

La siguiente figura, muestra un esquema de cómo queda el trabajo con sus conexiones entre hardware y que parte del trabajo hace cada uno. Además de tener en cuenta el voltaje y corriente que hay que suministrarle a cada dispositivo.

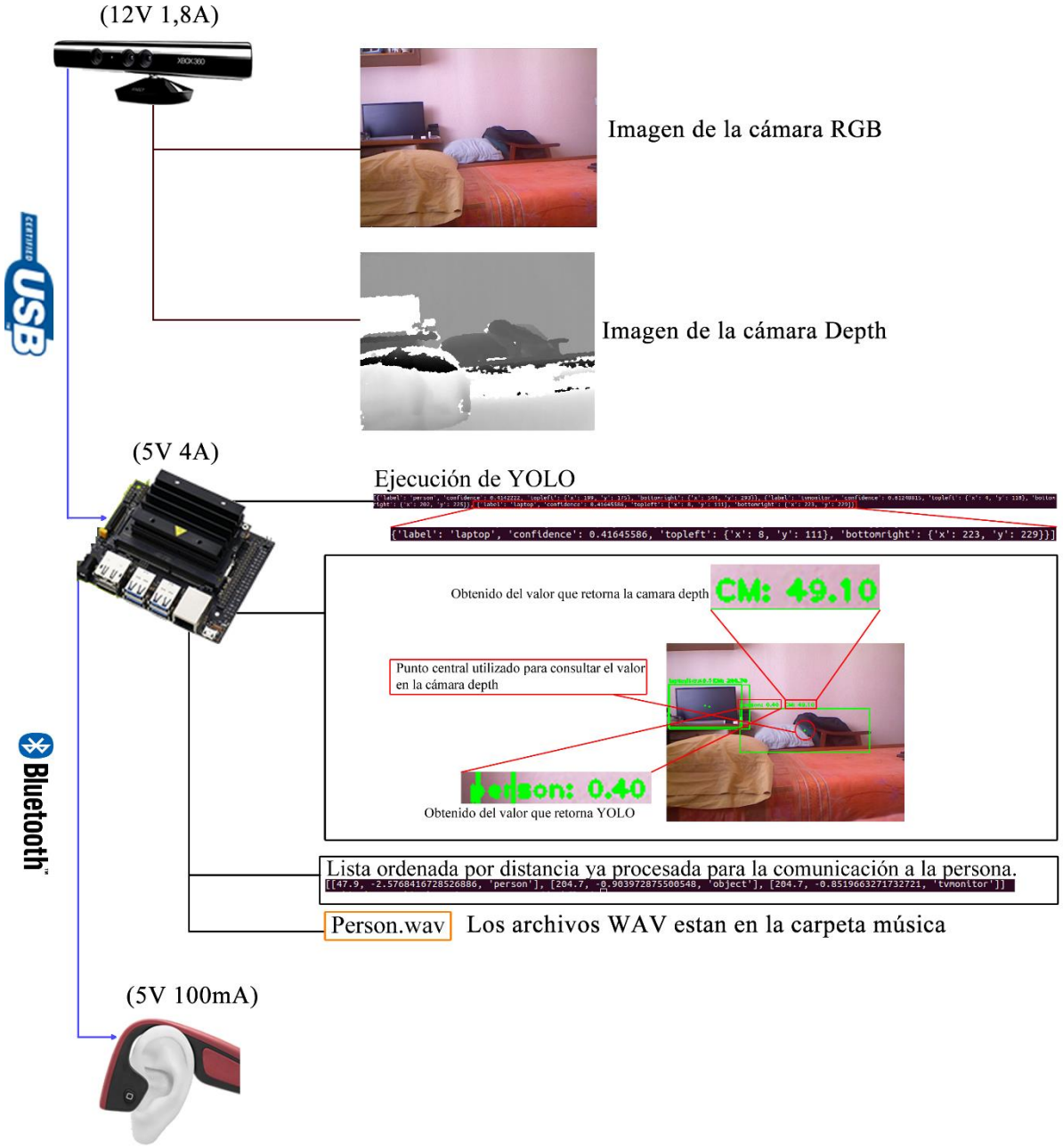

Figura 47: Esquema de las interconexiones de los distintos hardware con su parte de código ejecutada.

Conclusiones y cuestiones abiertas

Este proyecto ha sido todo un reto a la hora de implementar todas las pruebas que he realizado, ya que, según las configuraciones, obtenía uno u otro tipo de feedback.

Inicialmente la idea era presentar algo como prototipo final, pero desde un principio ha ido mutando y se ha convertido en un trabajo de desarrollo de investigación sobre cuáles son los materiales que resultarían mejor en un prototipo final.

He desarrollado distintos softwares para cada una de las pruebas en lenguaje Python, el cual me ha resultado muy agradable para este tipo de tareas y bastante cómodo a la hora de manejar ventanas con opencv.

Las pruebas realizadas han dado resultados sorprendentes, la idea principal era trabajar con la raspberry pi 4, pero a medida que iba utilizándola se veía que su capacidad de procesar imágenes con YOLO era muy limitada.

En este interín hemos descubierto con la ayuda de mi tutor distintos miniordenadores, entre los cuales están la Jetson Nano de NVIDIA, que permiten hacer el trabajo de forma más eficiente. Si bien el mejor rendimiento lo obtenemos con el ordenador portátil, con la Jetson Nano la fluidez es aceptable como para resolver problemas en tiempo real y su portabilidad es excelente.

En cuanto a la fase de procesamiento es la que más queda por desarrollar, ya que el tipo de selección del orden de los objetos a nombrar no es la forma más inteligente de comunicarse con el usuario.

Se podría probar a futuro la cámara de Intel Realsense, que promete mejores resultados a la hora de obtener imágenes más nítidas, mejor portabilidad y velocidad de respuesta.

El punto anterior es de investigación, ya que no sé, si es por YOLO o por la cámara Kinect, pero realizando pruebas con imágenes puedo darme cuenta de que las detecciones y los bounding boxes que dibuja YOLO no son los que me esperaba por completo.

Para la fase de procesamiento también sería interesante detectar los objetos que se mueven hacia nosotros o que se van de nosotros y a qué velocidad. Para ello podríamos utilizar un sensor Lidar como los que llevan los coches autónomos, ya que nos escanean un mapa de distancias con una precisión muy grande y podemos ver también a la velocidad que esas distancias se acortan o aumentan. Ya sabemos que los peligros de la calle no sólo son los que tiene el usuario frente a sus ojos, sino que puede haber peligros móviles que vengan desde atrás como un ciclista distraído, etc.

Otro aspecto importante para agregar es la detección de desniveles y la comunicación que hay que darle al usuario sobre estos. Conocemos bien que el bastón de las personas invidentes o con pérdida de visión se utiliza para detectar objetos, pero además detectar desniveles. Avanzar y que el bastón no encuentre resistencia en el suelo es signo de que hay un escalón o desnivel.

Podríamos intentar comunicar este tipo de situaciones mediante vibración en las manos más que mediante sonidos, o probar un hibrido de los dos. Existe un proyecto muy interesante de Microsoft [8] que me ha mostrado mi tutor que se trata de un sistema que lleva la realidad virtual a las personas invidentes. Los conceptos son muy interesantes como para tomarlos de punto de partida para poder adaptarlos a nuestro proyecto.

Otro punto interesante a seguir refinando es la forma de comunicarle al usuario los objetos detectados cuando hay muchos en la imagen. He dejado que sólo nombre los tres mas precisos que ha podido detectar, pero aun así no me parece que sea la opción mas adecuada.

Sería interesante para este tipo de cuestiones poder realizar un modelo de pruebas con el cuál salir a la calle con una persona invidente y que nos de feedback sobre que es lo que realmente está intentando obtener de información en cada momento. Esto nos permitiría poder ajustar mas la forma de la comunicación con el usuario y avanzar de esta manera hacia un modelo más eficiente.

Referencias

[1] Michal Bujacz,Karol Kropidlowski, Gabriel Ivanca, Alin Moldoveanu, Charalampos Saitis, Adam Csapo, György Wersenyi, Simone Spagnil, Omar I. Johannesson, Runar Unnthorsson, Mikolai Rotnicki, Piotr Witek (2016). Sound of Vision.

[2] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi (2016). You Only Look Once: Unified, Real-Time Object Detection. [Figura 2]. Recuperado de https://arxiv.org/pdf/1506.02640.pdf

[3] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi (2016). You Only Look Once: Unified, Real-Time Object Detection. [Figura 3]. Recuperado de https://arxiv.org/pdf/1506.02640.pdf

[4] Bosun Xie (2013). Head-related Transfer Function and Virtual Auditory Display: Second Edition. [Figura]. Figure 1.14 Diagram of sound transmission from a point sound source to the two ears.

[5] Bosun Xie (2013). Head-related Transfer Function and Virtual Auditory Display: Second Edition. [Figura]. Figure 1.15 Models of sound transmission through the human external ear: (a) anatomikal sketch; (b) analogue model (adapted from Møller, 1992)

[6] Libfreenect. Librería Python para el control de la cámara Kinect. Recuperado de https://openkinect.org/wiki/Main_Page

[7] Libaudioverse. Librería para la gestión de audio en 3D para Python. Recuperado de https://libaudioverse.github.io/libaudioverse/docs/branches/master/libaudioverse_manual.html

[8] Edward Cutrell, Eyla Ofek , Alexa F. Siu, Mike Sinclair, Robert Kovacs, Christian Holz(2020). Virtual Reality Without Vision: A Haptic and Auditory White Cane to Navigate Complex Virtual Worlds. Recuperado de https://www.microsoft.com/en-us/research/uploads/prod/2020/02/Siu-VR_without_vision-CHI2020.pdf

[9] Bosun Xie (2013). Head-related Transfer Function and Virtual Auditory Display: Second Edition

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi (2016). You Only Look Once: Unified, Real-Time Object Detection

[11] Peter Wlodarczak (2020). Machine learning and its applications. CRC Press/Taylor & Francis Group